Comment s'assurer de la qualité ?

Revue de code

Règles d'intégration

Importance des cas tests

Qualification de code

EDF R&D

Comment s'assurer de la qualité ?

Validation élémentaire - revue de code :

En continu et à échéances fixées

- Exécutions de la base de cas tests sur plusieurs plate-formes (portages).
- Code_Coverage. Un bloc de programmation où aucun test ne rentre a **100%** de chances de contenir un bug.
- Les cas tests couvrent-ils tous les usages possibles ? Les associations possibles (élément fini / matériau / loi de comportement) ?

Comment s'assurer de la qualité ?

Validation élémentaire – revue de code (suite) :

En continu et à échéances fixées

- Suivi des performances « en situation » : mise au point de cas tests non validants, plus simples qu'une étude, dont le seul but est de servir d'étalon à la mesure de performances.
- Suivi des performances individuelles des cas tests à chaque release
- Les messages d'erreur sont-ils tous intelligibles ?
- Relecture croisée des développements XP

Y travailler tous et tout le temps

Investir un responsable pour les tâches récurrentes

EDF R&D

Comment s'assurer de la qualité ?

Vérification de la qualité d'un développement lors de la phase d'intégration

Les réunions d'intégration sont structurées par les outils de vérification du développement :

- REX : cycle de vie des fiches
- Pré-inscription : vérification de règles de programmation
- Pré-inscription : rédaction d'une description synthétique
- Pré-inscription : compilation et exécution des nouveaux cas-tests
- Pré-inscription : cohérence de version et gestion des conflits de notation
- Livraison : idem + exécution de la base complète de cas-tests

EDF R&D

Comment s'assurer de la qualité ?

Vérification de la qualité d'un développement lors de la phase d'intégration

Règles de programmation :

- Normalisation de l'interface utilisateur
- Normalisation du source (on pallie le laxisme des compilateurs Fortran) : règles de nommage, limitation du nombre d'arguments et de la longueur des routines, instructions interdites (ou d'usage à justifier), contrôle du type des arguments
- Utilisation systématique d'outils logiciels partagés : routines utilitaires, mécanismes d'arrêt du code, d'envoi de messages, de lecture/écriture.

Comment s'assurer de la qualité ?

Importance des cas-tests. À quoi servent-ils ?

1. À valider un développement par comparaison à un **problème de référence** :
   - une solution analytique existe ou peut être développée
   - une solution numérique est obtenue par un autre code de calcul
   - on dispose de données expérimentales : tolérance souvent assez grande, doubler par un test de non régression
   - le problème peut être traité par un autre cheminement du logiciel : intra-comparaison
   - test de non régression (en dernier recours !)

Comment s'assurer de la qualité ?

Importance des cas-tests. À quoi servent-ils ?

② À servir de base représentative et exhaustive de l'usage du code :

- Pour la **revue de code**
- Pour la qualification des **portages** et installations locales

③ À fournir une base de tutoriaux aux utilisateurs

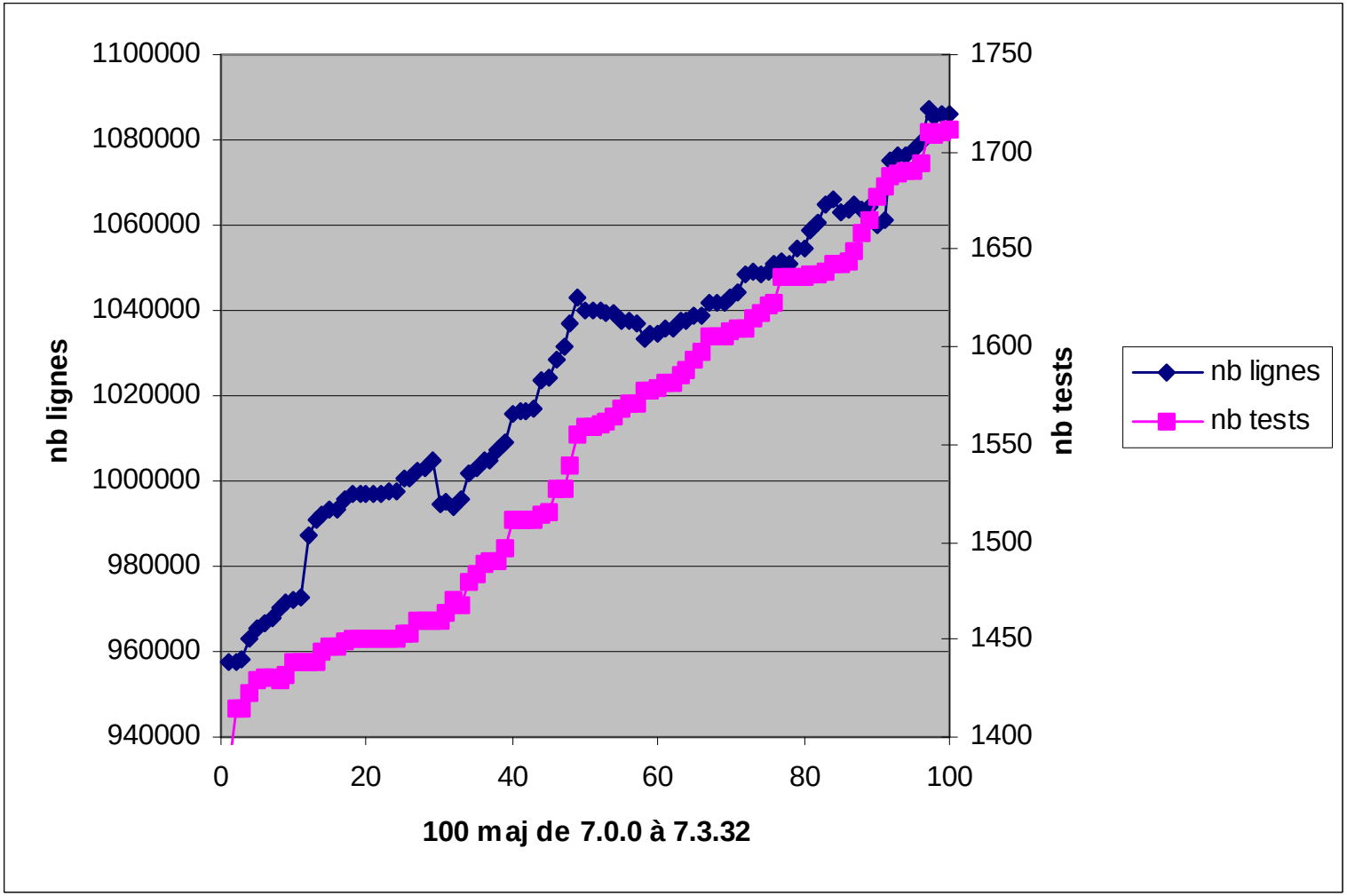
➡ Un cas test n'est pas le c/c d'une étude

➡ Un cas test doit être simple : analytique, intelligible

➡ Un cas test doit être documenté

Quelle organisation pour développer ?

Importance de la production continue de cas tests :



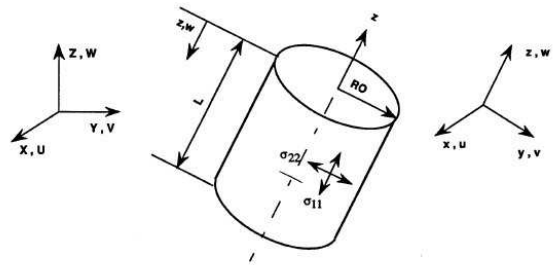
Quelle organisation pour développer ?

Code_Aster® Version 5.0

Titre : SSLV307 - Cylindre oblique sous charge axiale uniforme Date : 23/09/02
Auteur(s) : X. DESROCHES Clé : V3.04.307-A Page : 2/8

1 Problème de référence

1.1 Géométrie



Rayon moyen : R_o = 1 m
Epaisseur : h = 0.02 m
Hauteur : L = 4 m

Cosinus directeurs de l'axe du cylindre : $(0.0 , 0,5 , \frac{\sqrt{3}}{2})$

Axe local x parallèle à l'axe global X.

1.2 Propriétés de matériaux

E = 2.1 x 10^{11} Pa

v = 0.3

1.3 Conditions aux limites et chargements

- Déplacement axial nul à l'extrémité basse (w = 0)
  Pour les autres conditions aux limites (relations linéaires), voir paragraphe [§3].
- Charge axial uniforme par unité de longueur q = 10000 N/m, appliquée à l'extrémité haute.

1.4 Conditions initiales

Sans objet pour l'analyse statique.

Manuel de Validation Fascicule V3.04 : Statique linéaire des structures volumiques HT-66/02/001/A

Structuration d'une documentation de cas tests

Toutes infos utiles pour rejouer la scène … ou benchmarker avec un autre code !

Géométrie, matériaux, conditions aux limites, modèles employés

Quelle organisation pour développer ?

Code_Aster® Version 5.0

Titre : SSLV307 - Cylindre oblique sous charge axiale uniforme Date : 23/09/02
Auteur(s) : X. DESROCHES Clé : V3.04.307-A Page : 3/8

2 Solution de référence

2.1 Méthode de calcul utilisée pour la solution de référence

- Déplacement radial en repère local (x, y, z) :

$$u_r = \frac{qvRo}{Eh} = -\left[U^2 + \left(\frac{\sqrt{3}}{2}V - 0.5\,W\right)^2\right]^{1/2}$$

où U, V, W = composantes du déplacement dans le repère global (X, Y, Z).

- Si $\sigma_{xx}, \sigma_{yy}\ \sigma_{zz} = \sigma_{11}$ sont les contraintes dans le repère local, les contraintes exprimées dans le repère global valent :

$$\sigma_{xx} = \sigma_{xx}$$
$$= 3/4\sigma_{yy} + 1/4\sigma_{11} \qquad \sigma_{11} = q/h$$
$$\sigma_{zz} = 1/4\sigma_{yy} + 3/4\sigma_{11}$$
$$\sigma_{yz} = -\frac{\sqrt{3}}{4}\sigma_{yy} + \frac{\sqrt{3}}{4}\sigma_{11}$$

Dans le plan local (x, z), $\sigma_{yy} = 0$ (contrainte circonférentielle),

d'où $\sigma_{yy} 1/4\ \sigma_{11}, \sigma_{zz} = 3/4\ \sigma_{11}$

2.2 Résultats de référence

- Déplacement raidal : u_r = –7.14 x 10^{-7} m
- Dans le plan local (x, z), σ_{yy} = 1.25 x 10^5 Pa, σ_{zz} = 3.75 x 10^5 Pa

2.3 Incertitude sur la solution

- Solution analytique

2.4 Références bibliographiques

[1] R. J. ROARK et W. C. YOUNG : Formulas for stress and strain, 5e édition. New-York, Mc Graw-Hill, 1975

Manuel de Validation Fascicule V3.04 : Statique linéaire des structures volumiques HT-66/02/001/A

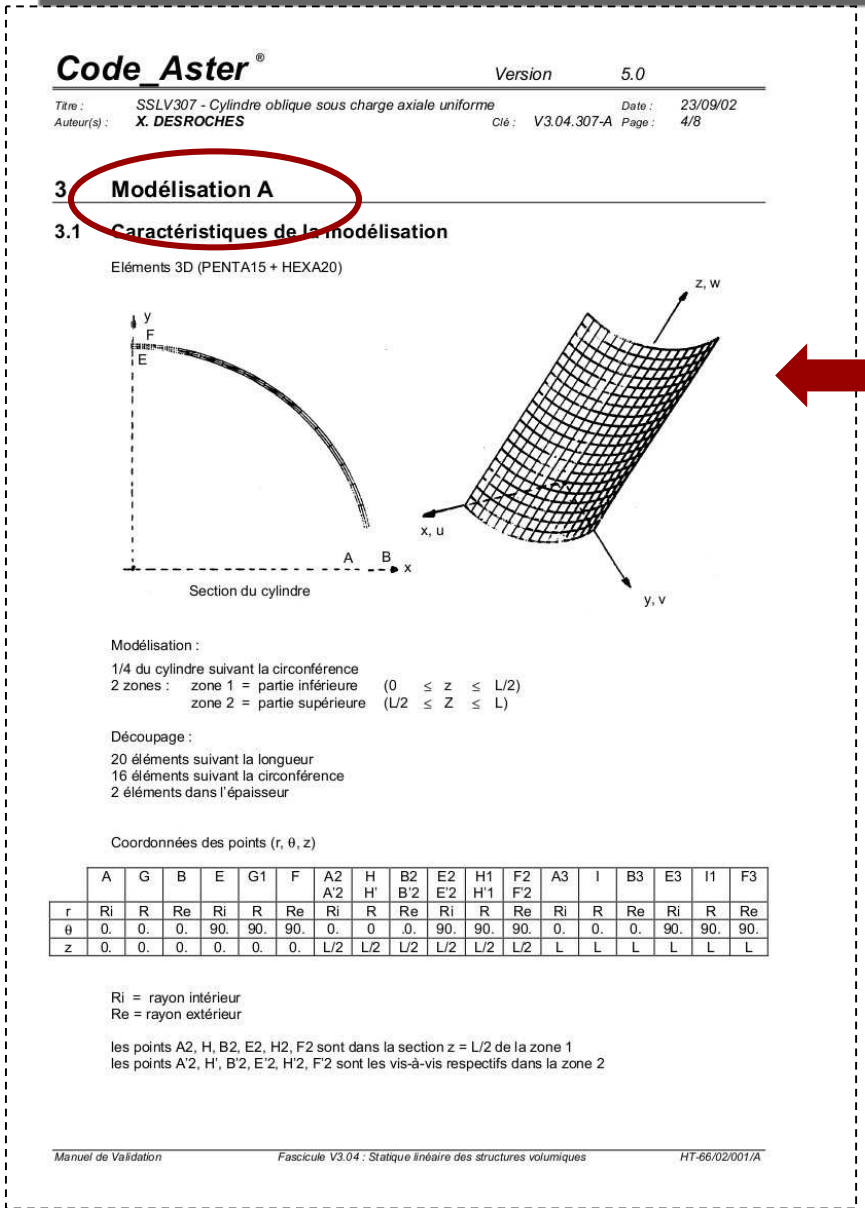
Structuration d'une documentation de cas tests

Description de la solution de référence

On précise la valeur numérique testée

On cite les sources

Quelle organisation pour développer ?



Code_Aster® Version 5.0

Titre : SSLV307 - Cylindre oblique sous charge axiale uniforme Date : 23/09/02
Auteur(s) : X. DESROCHES Clé : V3.04.307-A Page : 4/8

3 Modélisation A

3.1 Caractéristiques de la modélisation

Eléments 3D (PENTA15 + HEXA20)

Modélisation :

1/4 du cylindre suivant la circonférence
2 zones : zone 1 = partie inférieure (0 ≤ z ≤ L/2)
zone 2 = partie supérieure (L/2 ≤ Z ≤ L)

Découpage :

20 éléments suivant la longueur
16 éléments suivant la circonférence
2 éléments dans l'épaisseur

Coordonnées des points (r, θ, z)

	A	G	B	E	G1	F	A2 A'2	H H'	B2 B'2	E2 E'2	H1 H'1	F2 F'2	A3	I	B3	E3	I1	F3
r	Ri	R	Re	Ri	R	Re	Ri	R	Re	Ri	R	Re	Ri	R	Re	Ri	R	Re
θ	0.	0.	0.	90.	90.	90.	0.	0	.0.	90.	90.	90.	0.	0.	0.	90.	90.	90.
z	0.	0.	0.	0.	0.	0.	L/2	L/2	L/2	L/2	L/2	L/2	L	L	L	L	L	L

Ri = rayon intérieur
Re = rayon extérieur

les points A2, H, B2, E2, H2, F2 sont dans la section z = L/2 de la zone 1
les points A'2, H', B'2, E'2, H'2, F'2 sont les vis-à-vis respectifs dans la zone 2

Manuel de Validation Fascicule V3.04 : Statique linéaire des structures volumiques HT-66/02/001/A

Structuration d'une documentation de cas tests

Description de la modélisation

La comparaison numérique calcul / référence est faite dans le test lui-même avec procédure d'alarme en cas d'écart.

Comment s'assurer de la qualité ?

Qualification de code

Bien suivre les TP de formation :

- Un bizuth a toutes les chances de « solliciter » votre code dans des usages inhabituels quand l'utilisateur expérimenté rejouera tout le temps la même scène.
- Les TP sont souvent source de mise à jour de bug de robustesse, de défaut de documentation, de réflexion sur l'ergonomie de votre logiciel.

Sensibiliser les utilisateurs sur l'importance du **FEEDBACK**

Via le REX, via le club Utilisateurs

Qualification de code

Le logiciel est-il apte à simuler le problème de l'utilisateur ?

- Chaque unité utilisatrice valide le code pour son usage propre à une échelle globale (versus cas test : échelle fonctionnelle locale) : fonction « bureau des méthodes »
- On doit valider : la capacité à faire en configuration d'étude, la qualité des résultats par rapport au fonds ancien d'études, par rapport à des mesures sur site, par rapport à des benchmarks codes, par rapport à essais globaux.
- Ce travail est de la responsabilité des utilisateurs (bureaux d'études) : mais l'organisation code se doit de capitaliser toutes ces actions de qualification / benchmark pour les faire partager.

Comment diffuser ?

Panorama des services offerts à l'utilisateur :

Débutants :

- Formations
- Documentations « prise en main du code»
- Tutoriaux : cas tests

Tous utilisateurs :

- Corpus documentaire : Utilisation, Référence, Validation
- Docs utilisation : fonctions élémentaires + *howto* généraux
- Hotline
- Service d'assistance de niveau 2
- REX

Instances de représentation :

- Club Utilisateurs
- Structure de gouvernance

Comment diffuser ?

Victime de votre succès d'usage ?

Vous êtes facilement responsable de tout !

- Environnement informatique, administration serveur de calcul
- Utilisation et fonctionnement des outils périphériques

Vous devez aller à la pêche aux infos

- Sensibiliser l'utilisateur à la nécessité du feedback et lui offrir les outils pour ça : rex, club U
- L'encourager aux critiques (constructives) pour le rendre solidaire de l'amélioration du logiciel

Protégez-vous

- Des sollicitations excessives : aider à faire n'est pas faire
- Contractualisez les services par une organisation réactive mais rigoureuse

Comment diffuser ?

Mais n’oubliez non plus jamais que les utilisateurs sont l’unique finalité de votre travail !

Comment diffuser ?

Motivations pour diffuser en logiciel libre :

- Reconnaissance par l'usage

  Notoriété, comparaison à l'état de l'art

- Qualification

  Démultiplication de l'usage, benchmarks

- Contributions

  Complétude, documentation

- Diffusion des compétences

  Usages en école, université. Prestataires de services.

- Support à la construction de coopérations

  Linus Torvalds promeut le modèle libre par « l'efficacité de la coopération technique qu'il rend possible »

Comment diffuser ?

Le modèle libre est pertinent pour les entreprises :

- Efficacité par la « professionnalisation » de chacun
- Capacité d'assemblage (pas d'asservissement)
- Possible partage sans perte d'autonomie

EDF R&D

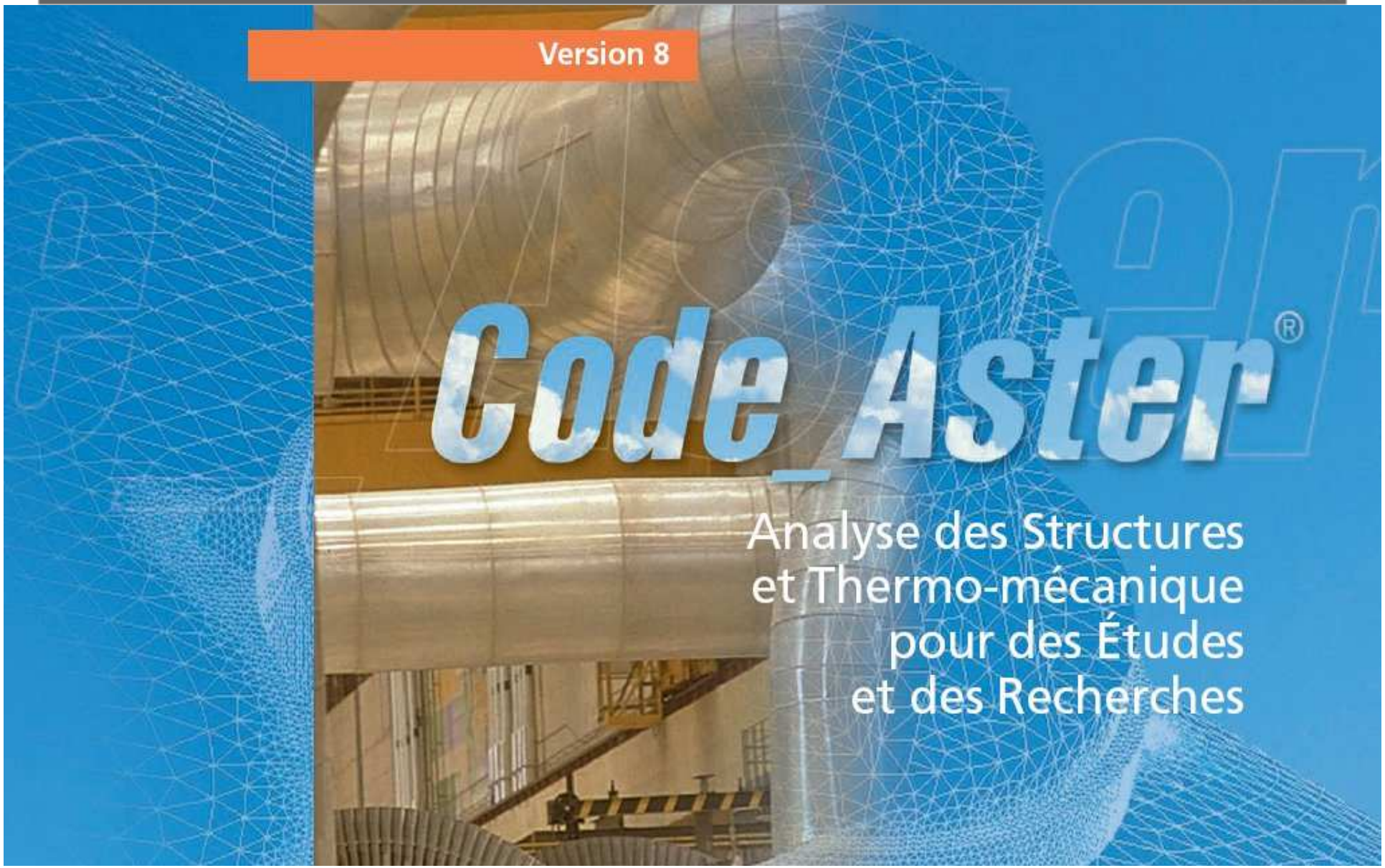
Conclusion

Les 6 qualités d'un bon logiciel :

- General purpose
- Reliable
- Free
- English
- Well managed
- Standard interface

[D. Nowak, Argonne National Laboratory]

Merci


Version 8
Code_Aster®
Analyse des Structures
et Thermo-mécanique
pour des Études
et des Recherches