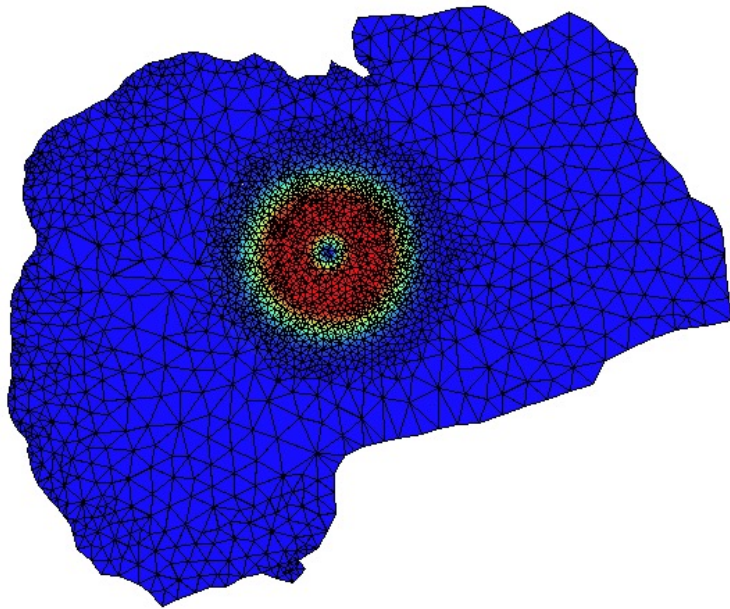


Reduced-gravity simulation of a baroclinic eddy in the Gulf of Mexico.

This simulation is several orders of magnitude cheaper than a constant resolution one of the same accuracy !

The Finite Element Method



Typical applications

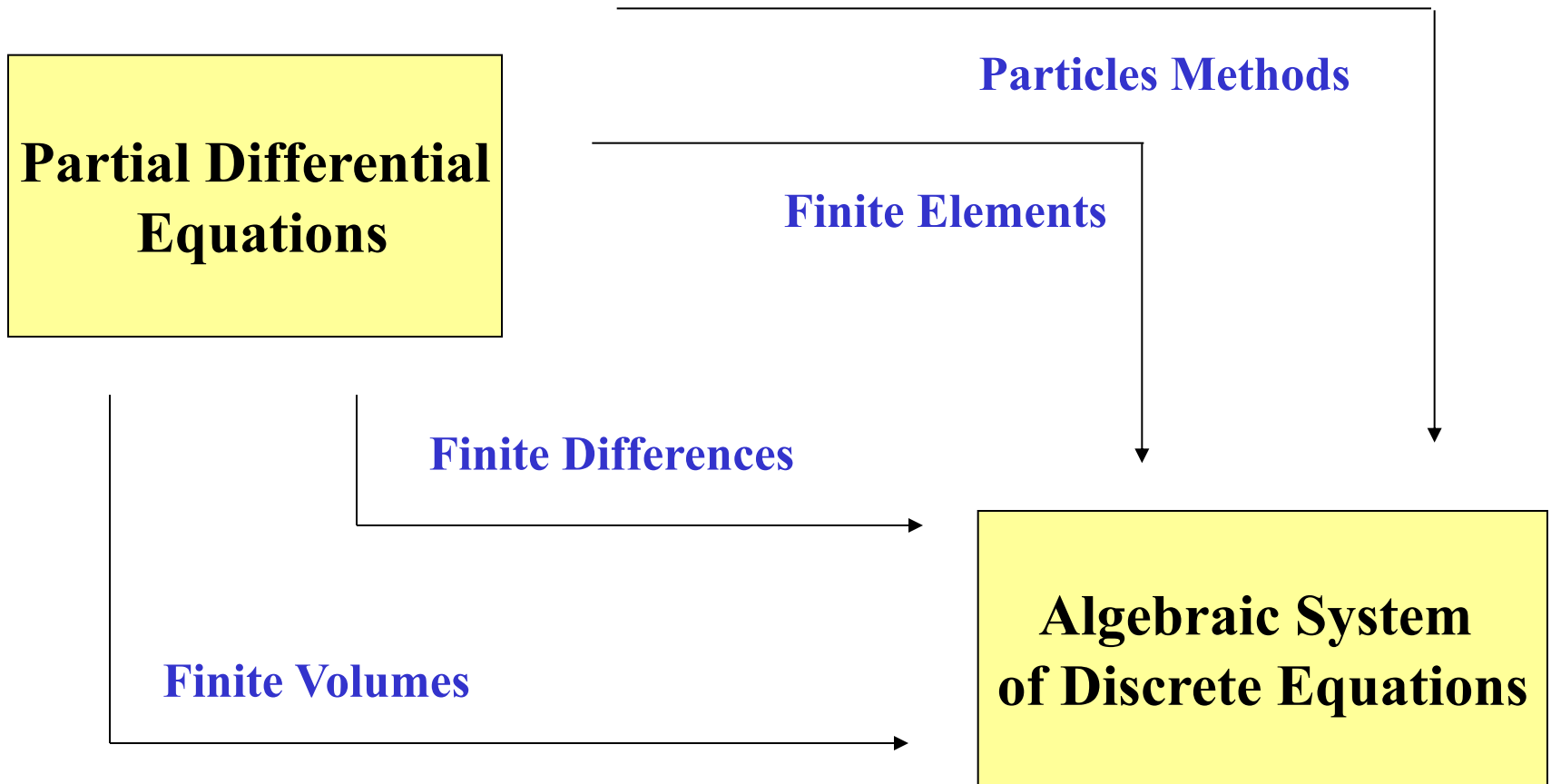
Deformable solids mechanics
Fluid dynamics (CFD)
Electromagnetism
Transport phenomena
Climatology

is a way of computing approximate solutions to a mathematical model describing a physical process.

What is a mathematical model ?
A boundary value problem.

What is a boundary value problem ?
A set of partial differential equations with boundary and initial conditions.

Finite Elements, Finite Differences, Finite Volumes etc.

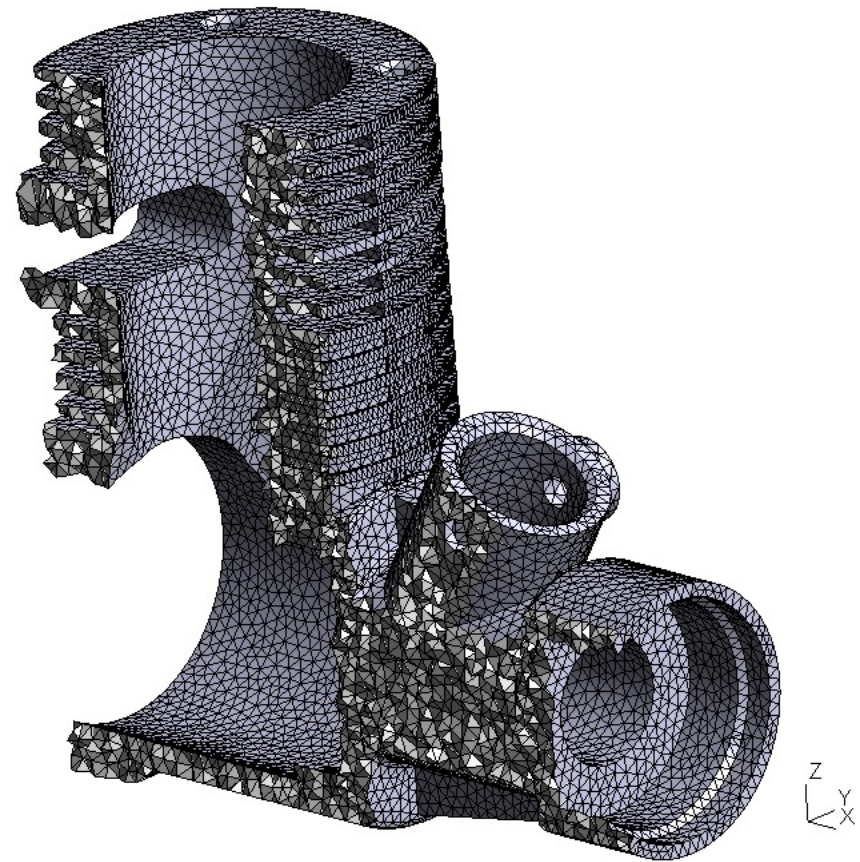


The Finite Elements Method is a discretization method

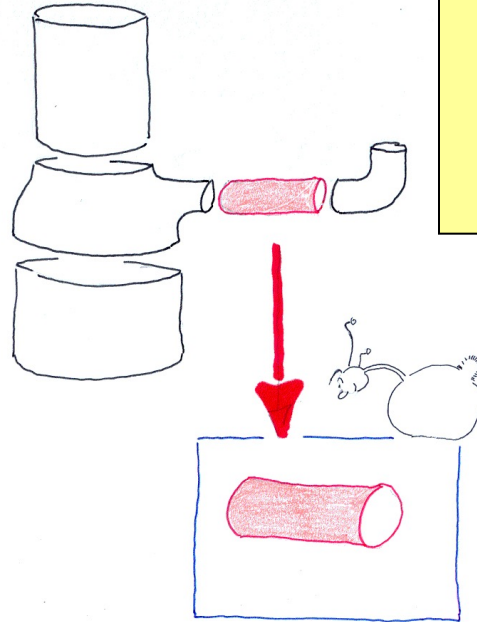
The problem geometry is divided in
small finite elements.

On each element, the solution is
approximated by means of unknown
nodal values and given polynomials

$$u(\mathbf{x}) \approx u^h(\mathbf{x}) = \sum_{j=1}^n U_j \tau_j(\mathbf{x})$$



Classical Engineering Analysis



**Exact solution
to approximate problems**

**Analysis through simple geometries
and a limited combination of
approximate models :**

**Lubrication theory
Bars
Beams
Plates and shells**

**Low computer's cost
Good physical understanding**

Simplicity of models

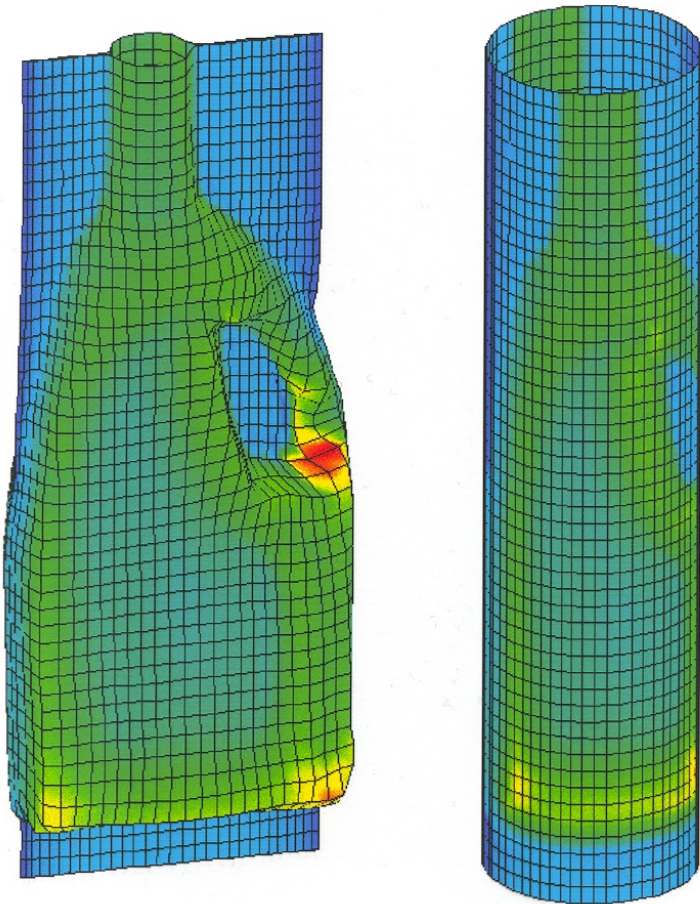
Complex geometries and loads cannot be handled

Complex materials cannot be analyzed

Computer Aided Engineering Analysis

The Finite Element Method provides approximate solutions to more realistic problems

FEM developed in the sixties for linear elasticity and generalized to many other applications...



Powerful and flexible

High (cheap) computer's cost

Low (expensive) engineer's cost

Complex processes can be analyzed

Complex material laws can be included

However...

Garbage in



Garbage out

Illusion of non-qualified users to be able to analyze everything

New modeling issues requires higher qualifications...

How to define complex problems in an accurate and efficient way for the computer software ?

A l'issue de ce cours, vous serez capables de...

- **Comprendre la méthode des éléments finis**
- **Réaliser un petit programme en C**
- **Certifier et valider une simulation**
- **Choisir la voie numérique la plus efficace**
- **Estimer la précision d'un résultat**
- **Découvrir les joies et les aléas du numérique**

Non, non : ceci on ne fera pas !

- **Apprendre le génie logiciel de l'orienté-objet**
- **Utiliser des logiciels commerciaux**
- **Faire de l'analyse numérique théorique**
- **Faire du calcul parallèle**
- **Résoudre les équations de Navier-Stokes**
- **Créer automatiquement des maillages**

Plan du cours et évaluation

Comment intégrer
numériquement
une fonction
sur un carré ?



A = Evaluation continue

S2-S10 : 8 cours et 8 petits problèmes **100%**

Comment prédire
un tsunami ?

B = Evaluation certificative

S2-S10 : séances d'exercices

S11-S13 : mini-projet **50%**

S11 : interrogation **50%**

En juin, note finale = $\min((A+B)/2, B)$

En septembre, examen oral + projet spécial

Evaluation

Objectifs du projet

Réaliser	Créer une application pour prédire un tsunami.
Certifier	Tester et valider le travail de votre groupe.
Expliquer	Expliquer de manière efficace et rapide à l'enseignant et aux autres étudiants ce que vous avez réalisé.
Comprendre	Comprendre ce que vous avez réalisé. Comprendre ce que d'autres groupes ont réalisé.

Exercices : 8 petits problèmes

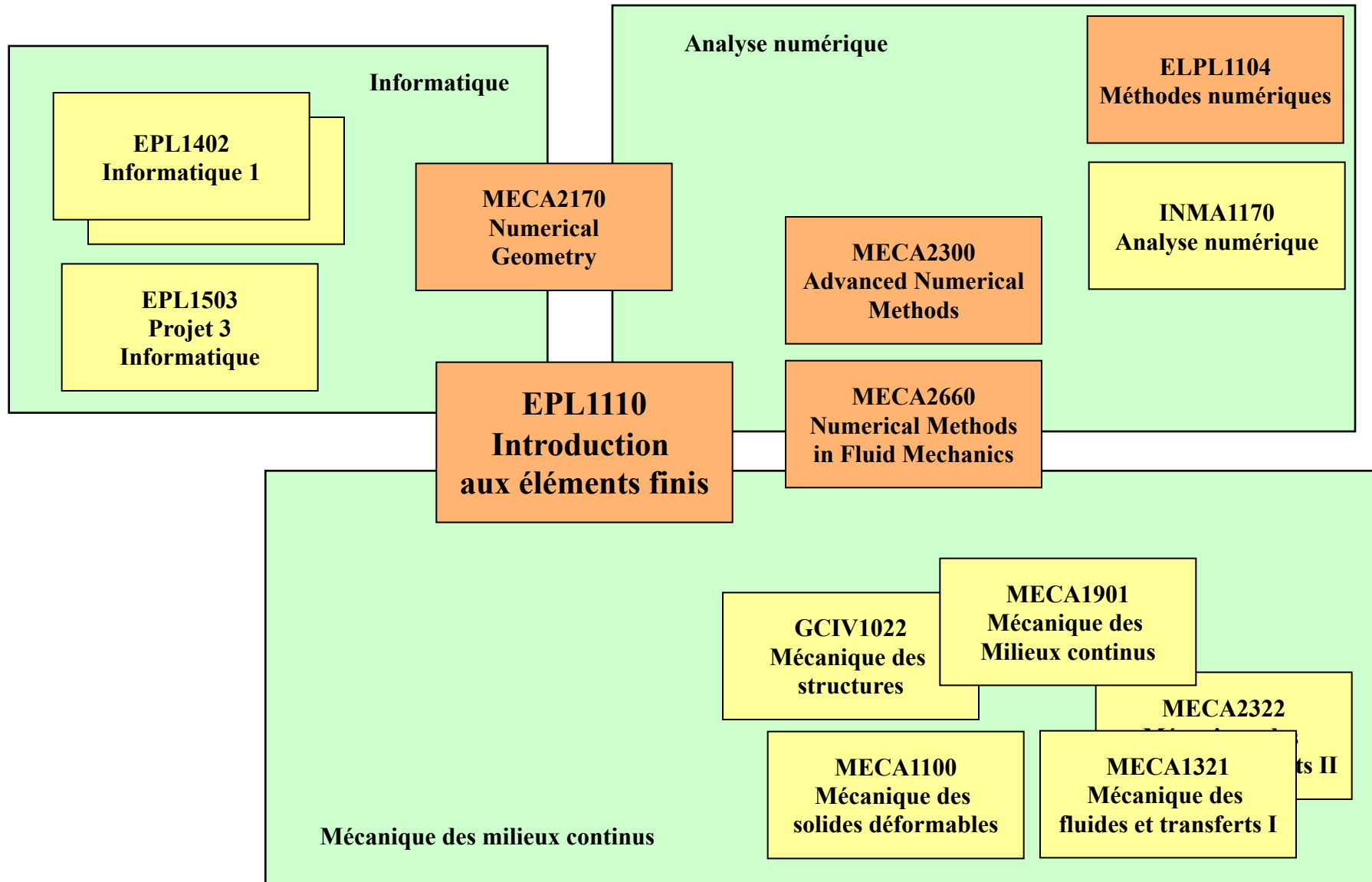
**Quelques petits problèmes
élémentaires pour apprivoiser le C**

Projet en C :

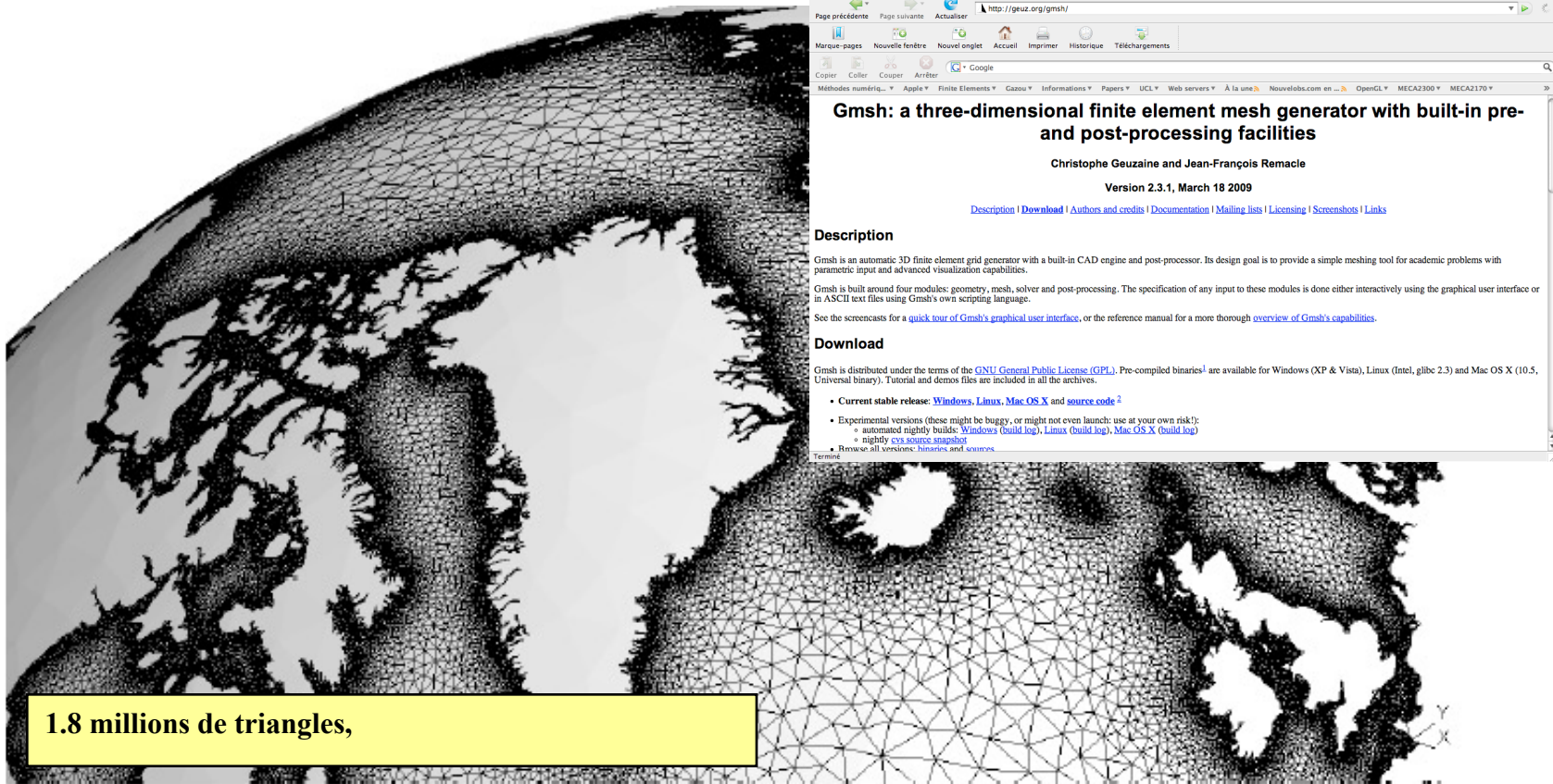
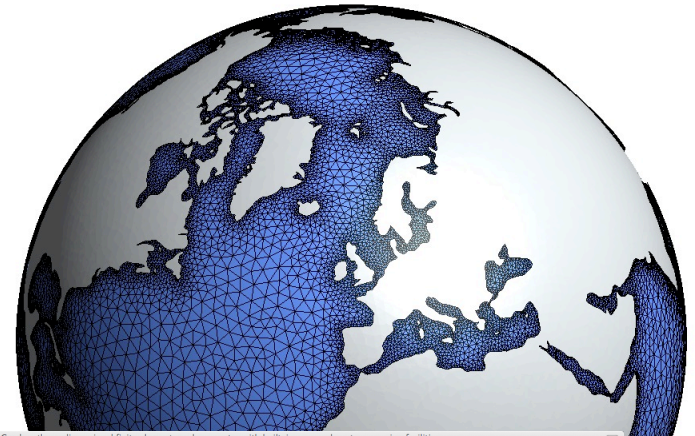
Une petite application efficace pour prédire un tsunami...

*The Practice of Programming :
Simplicity, Clarity, Generality. (B.K Kernighan & R. Pike 99)*

Et les autres cours....



Second-generation Louvain-la-Neuve Ice-ocean Model



Gmsh: a three-dimensional finite element mesh generator with built-in pre- and post-processing facilities

Page précédente Page suivante Actualiser
http://geuz.org/gmsh/

Marque-pages Nouvelle fenêtre Nouvel onglet Accueil Imprimer Historique Téléchargements

Méthodes numériq... Apple Finite Elements Cazou Informations Papers UCL Web servers À la une Nouvelobs.com en... OpenGL MECA2300 MECA2170

Copier Coller Couper Arrêter Google

Gmsh: a three-dimensional finite element mesh generator with built-in pre- and post-processing facilities

Christophe Geuzaine and Jean-François Remacle

Version 2.3.1, March 18 2009

[Description](#) | [Download](#) | [Authors and credits](#) | [Documentation](#) | [Mailing lists](#) | [Licensing](#) | [Screenshots](#) | [Links](#)

Description

Gmsh is an automatic 3D finite element grid generator with a built-in CAD engine and post-processor. Its design goal is to provide a simple meshing tool for academic problems with parametric input and advanced visualization capabilities.

Gmsh is built around four modules: geometry, mesh, solver and post-processing. The specification of any input to these modules is done either interactively using the graphical user interface or in ASCII text files using Gmsh's own scripting language.

See the screencasts for a [quick tour of Gmsh's graphical user interface](#), or the reference manual for a more thorough [overview of Gmsh's capabilities](#).

Download

Gmsh is distributed under the terms of the [GNU General Public License \(GPL\)](#). Pre-compiled binaries are available for Windows (XP & Vista), Linux (Intel, glibc 2.3) and Mac OS X (10.5, Universal binary). Tutorial and demos files are included in all the archives.

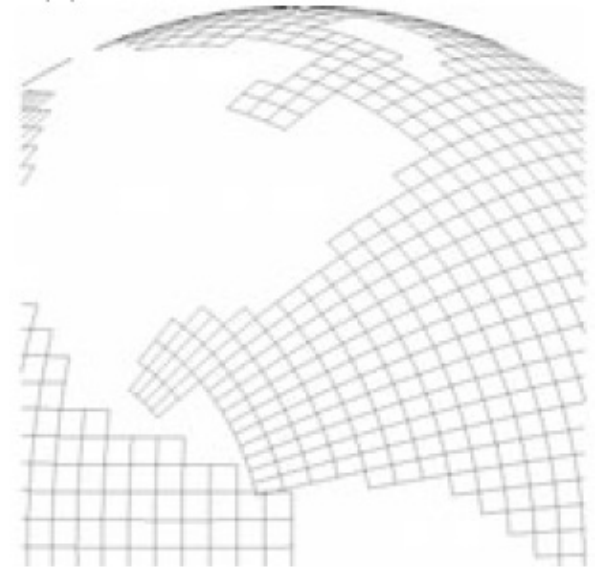
- Current stable release: [Windows](#), [Linux](#), [Mac OS X](#) and [source code](#) ²
- Experimental versions (these might be buggy, or might not even launch: use at your own risk!):
 - automated nightly builds: [Windows \(build log\)](#), [Linux \(build log\)](#), [Mac OS X \(build log\)](#)
 - nightly [cvs source snapshots](#)
- [Reverse all versions, histories, and sources](#)

Terminé

1.8 millions de triangles,

Structured grid ...

- **Finite differences are easy to implement**
- **Programming is easy**
- **Well known in the world of oceanography**
- **Bad representation of the coastlines**
- **Difficult to enhance locally the resolution**
- **Poles singularity**



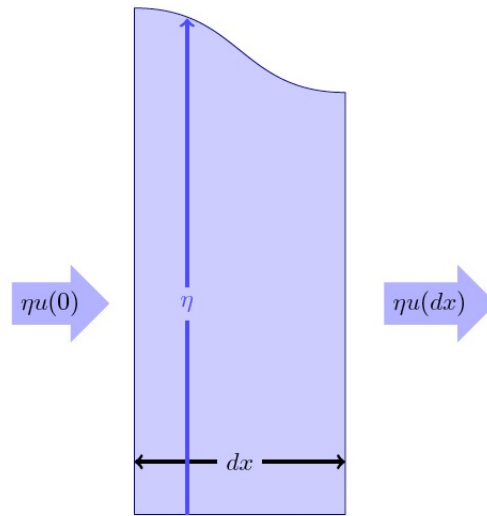
...versus unstructured grid



- **Numerical methods are more complicated**
- **Programming is more complicated**
- **Not well known in the world of oceanography**
- **Accurate representation of the coastlines**
- **Enhancing the resolution is flexible**
- **No singular points**

Mass balance

$$\frac{\partial \eta}{\partial t} + \eta_0 \frac{\partial u}{\partial x} = 0$$



$$dx \left(\rho \frac{\partial \eta}{\partial t} \right) = \rho \eta u(0) - \rho \eta u(dx)$$

$$\frac{\partial \eta}{\partial t} + \eta_0 \left(\frac{u(dx) - u(0)}{dx} \right) = 0$$



$$\frac{\partial \eta}{\partial t} + \eta_0 \frac{\partial u}{\partial x} = 0$$

$$dx \left(\rho \eta_0 \frac{\partial u}{\partial t} \right) = \rho g \frac{\eta^2(0)}{2} - \rho g \frac{\eta^2(dx)}{2}$$

$$\eta_0 \frac{\partial u}{\partial t} + g \eta_0 \frac{\partial \eta}{\partial x} = 0$$



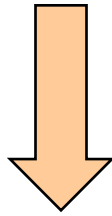
$$\frac{\partial u}{\partial t} + g \frac{\partial \eta}{\partial x} = 0$$

Momentum balance

$$\frac{\partial u}{\partial t} + g \frac{\partial \eta}{\partial x} = 0$$

$$\left\{ \begin{array}{l} \frac{1}{\eta_0} \frac{\partial \eta}{\partial t} + \frac{\partial u}{\partial x} = 0 \\ \frac{\partial u}{\partial t} + g \frac{\partial \eta}{\partial x} = 0 \end{array} \right.$$

**Linear
Shallow Water
Equations**



$$\frac{\partial^2 \eta}{\partial t^2} = g \eta_0 \frac{\partial^2 \eta}{\partial x^2}$$

Wave Equation

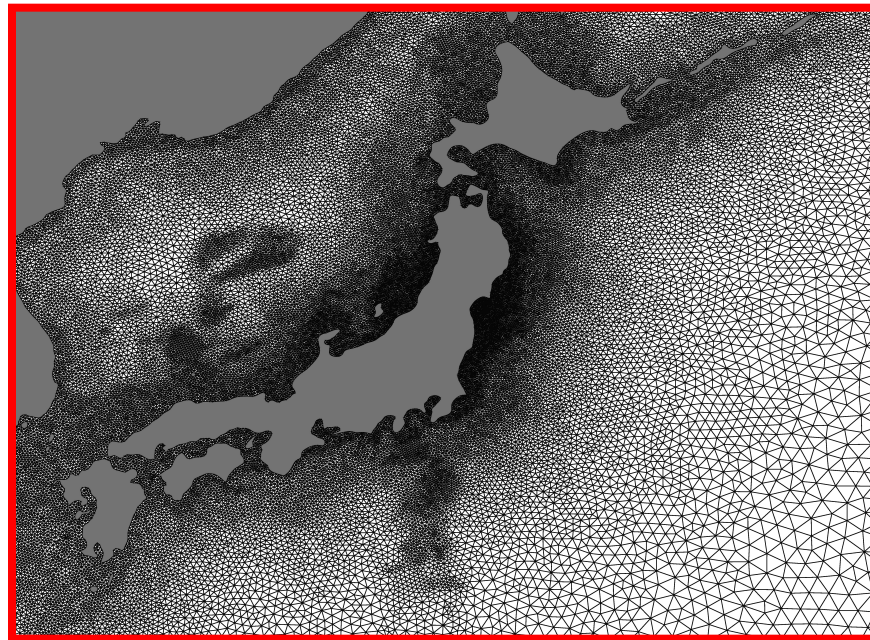
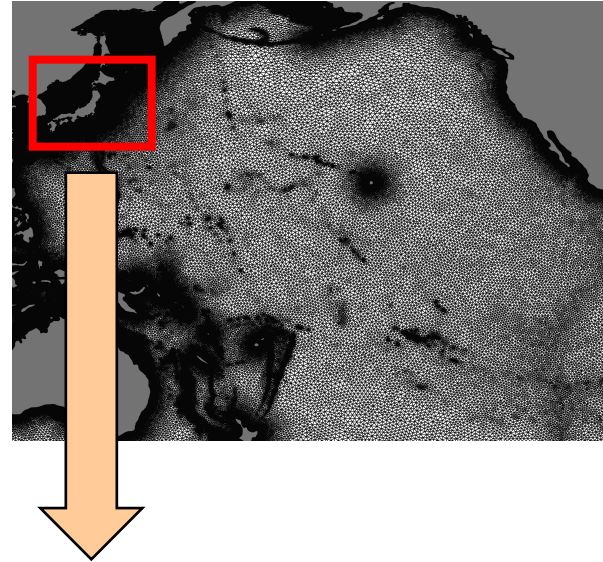
$$c = \sqrt{g \eta_0} \approx 200 \text{ [m/s]}$$

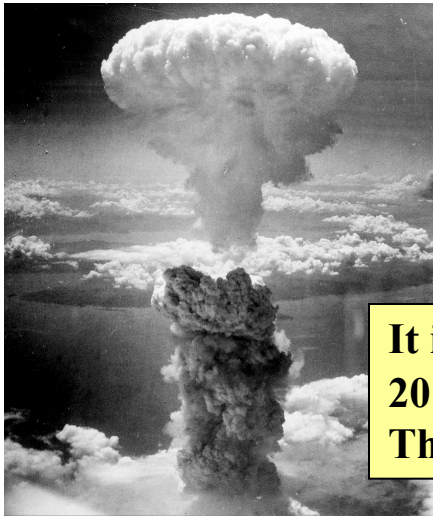
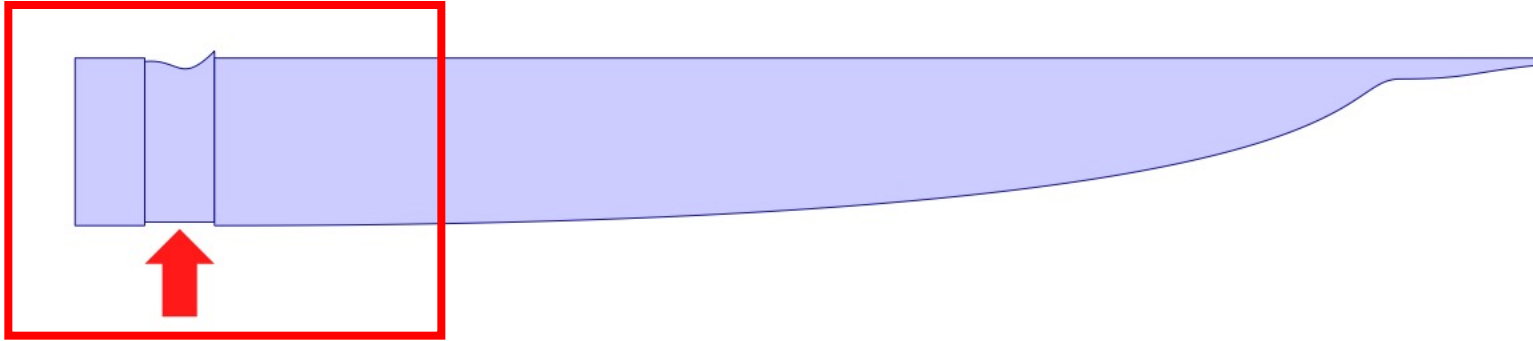
Gravity 9,81 m/s

Average depth of Pacific 4000 m

Waves are (very) fast !

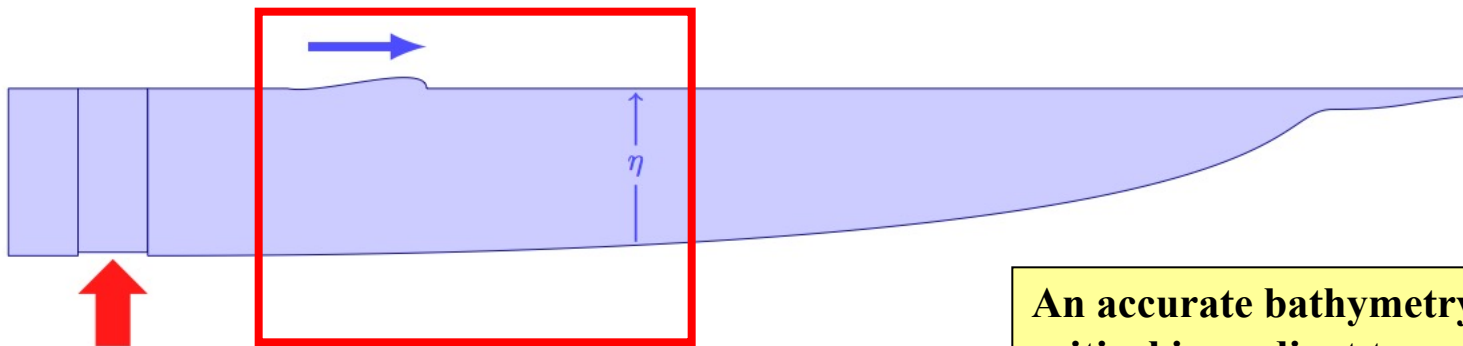
And now,
we can zoom
on Japan !





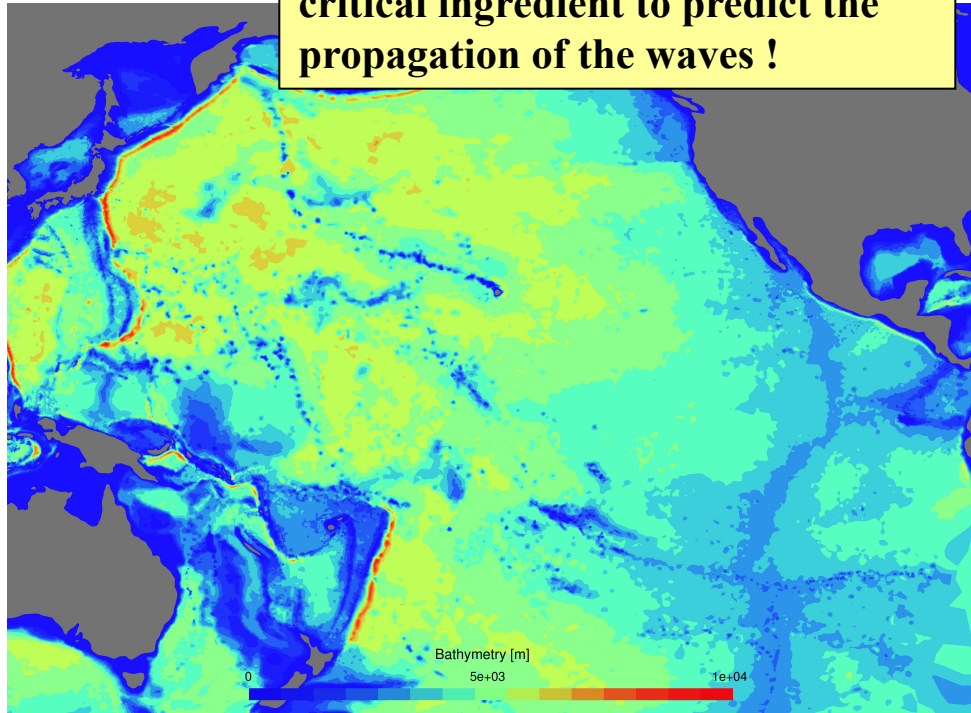
**It is a huge energy !
20 x Energy of Hiroshima's bomb !
This initial condition must be provided to the model**

**The earthquake motion
displaces a column water**

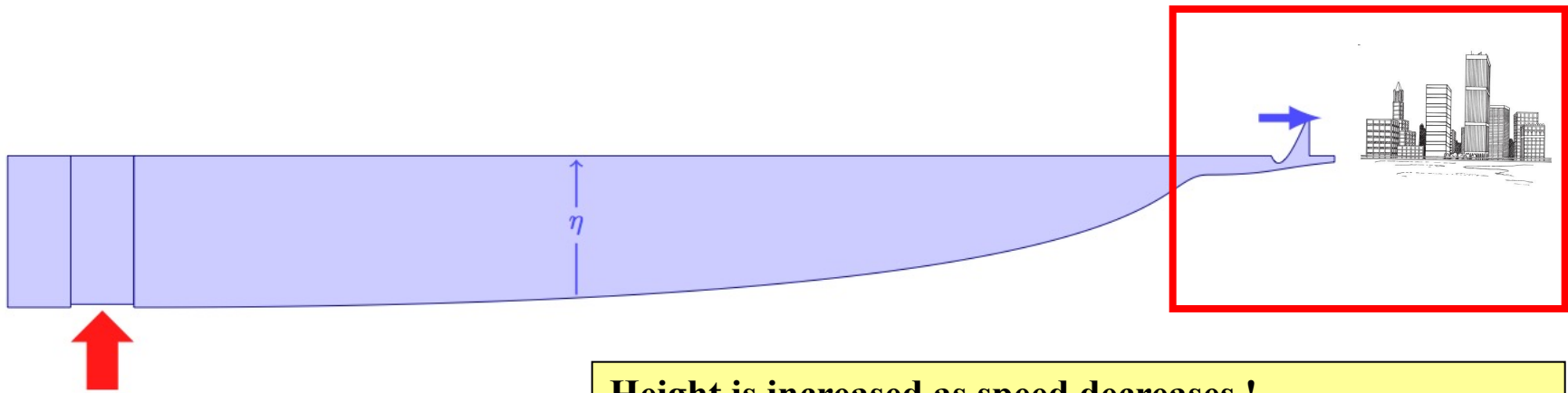


$$c = \sqrt{g\eta_0} \approx 200 \text{ [m/s]}$$

An accurate bathymetry is a critical ingredient to predict the propagation of the waves !



Small waves travel fast as function of the bathymetry



**Height is increased as speed decreases !
An accurate shoreline description is required.**



**Waves compression
forces waves to gain height !**

Simulating tsunamis is easy

Simulation performed on May 17th by Jonathan Lambrechts and Benjamin de Brye



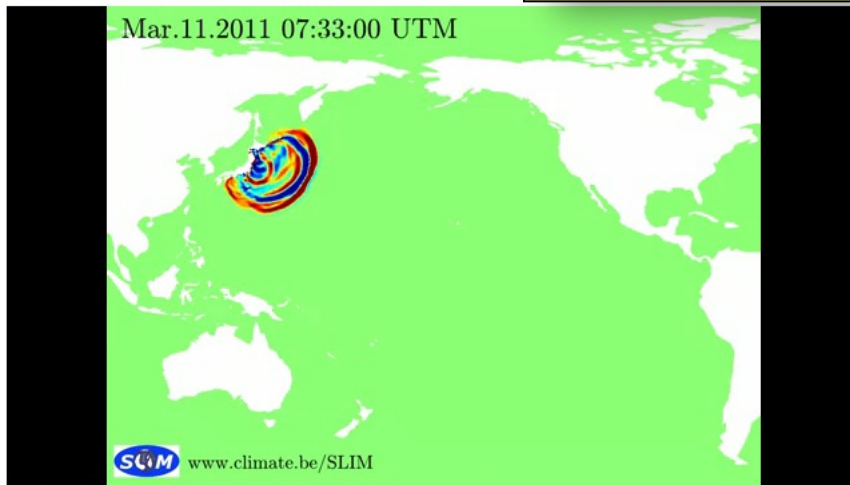
YouTube

Animation of March 11, 2011 Honshu tsunami

benjamindebrye

2 vidéos

S'abonner



0:02 / 0:34 360p

J'aime Ajouter à Partager

8 087

Ajoutée par benjamindebrye le 17 mars 2011

Propagation of the Honshu tsunami across the Pacific Ocean on March 11,

8 aime, 1 n'aime pas

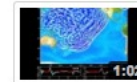
Plus

Suggestions



Animation of March 11, 2011 Honshu tsunami prop...

de benjamindebrye
597 vue(s)



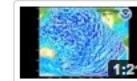
NOAA Animation of Tsunami Propagation from Earth...

de ExWeather
410 684 vue(s)



Ocean Floor Affects Japan Tsunami Propagation

de italk2youdotcom
29 031 vue(s)



Narrated animation of March 11, 2011 Honshu, J...

de NOAA PMEL
56 957 vue(s)



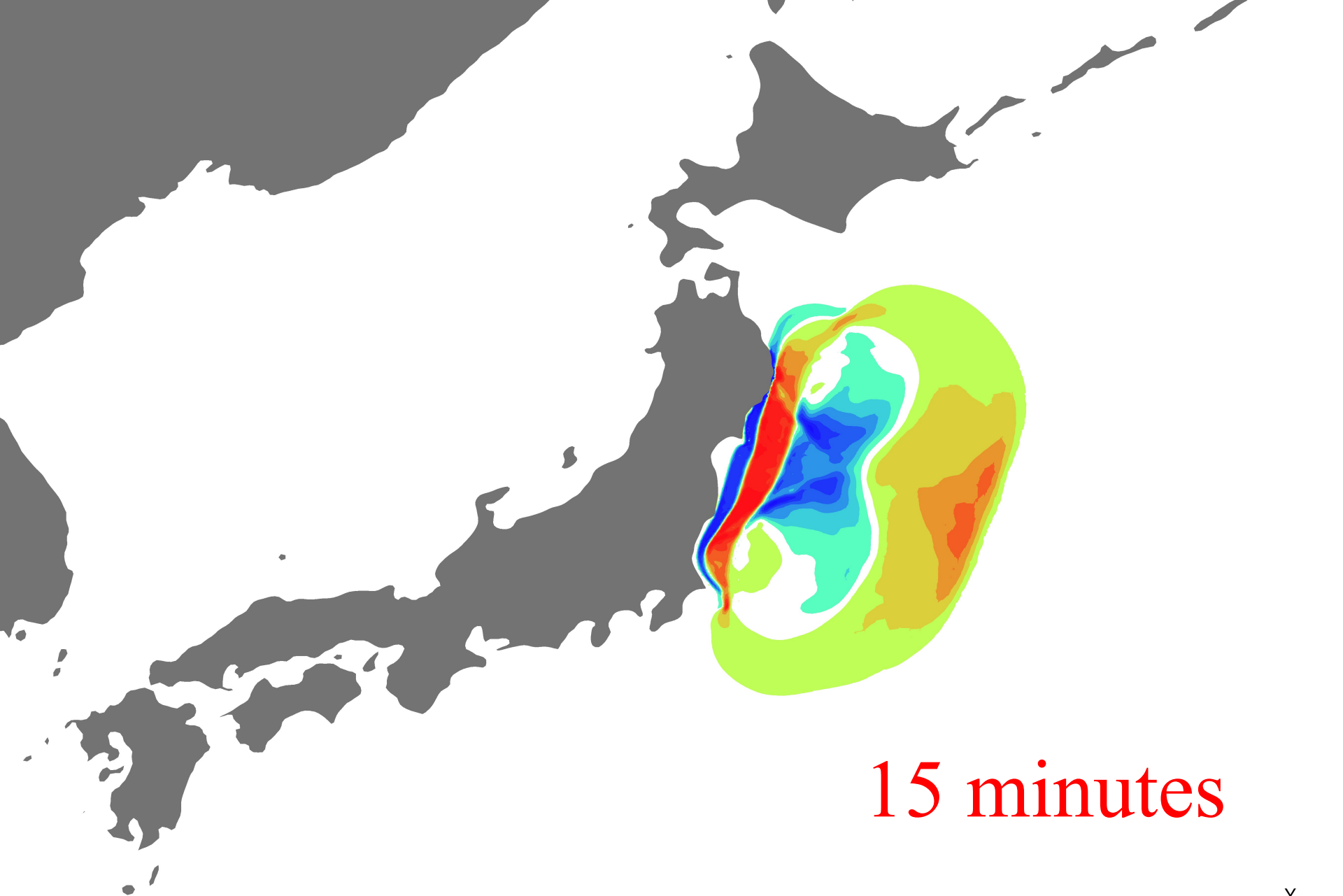
New Shocking rare video: Running From Tsunami i...

de japanquake2011
44 982 vue(s)



2011 Japan Sendai Tsunami Propagation 3D Simula...

de artisticoex
3 765 vue(s)

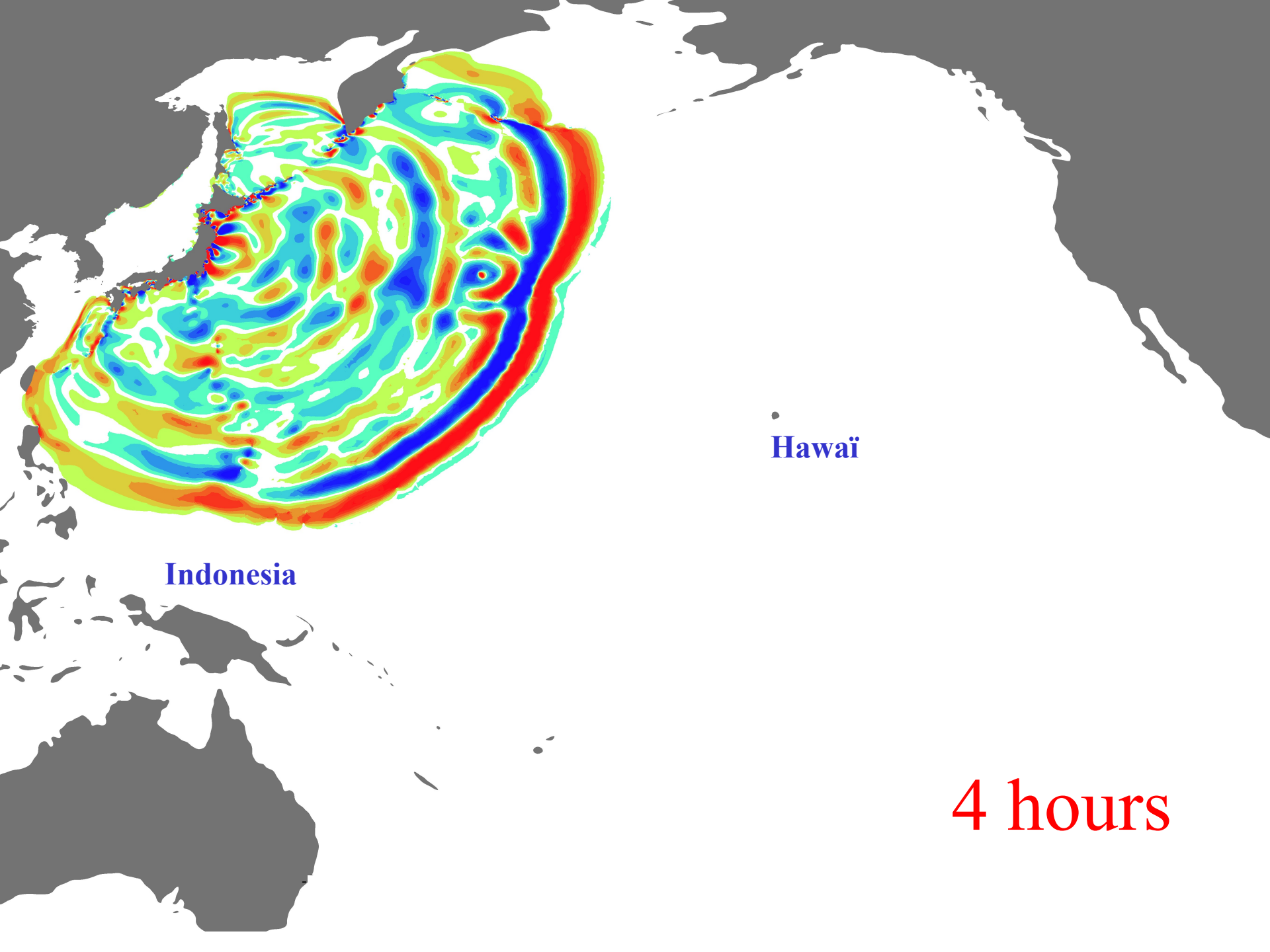


15 minutes





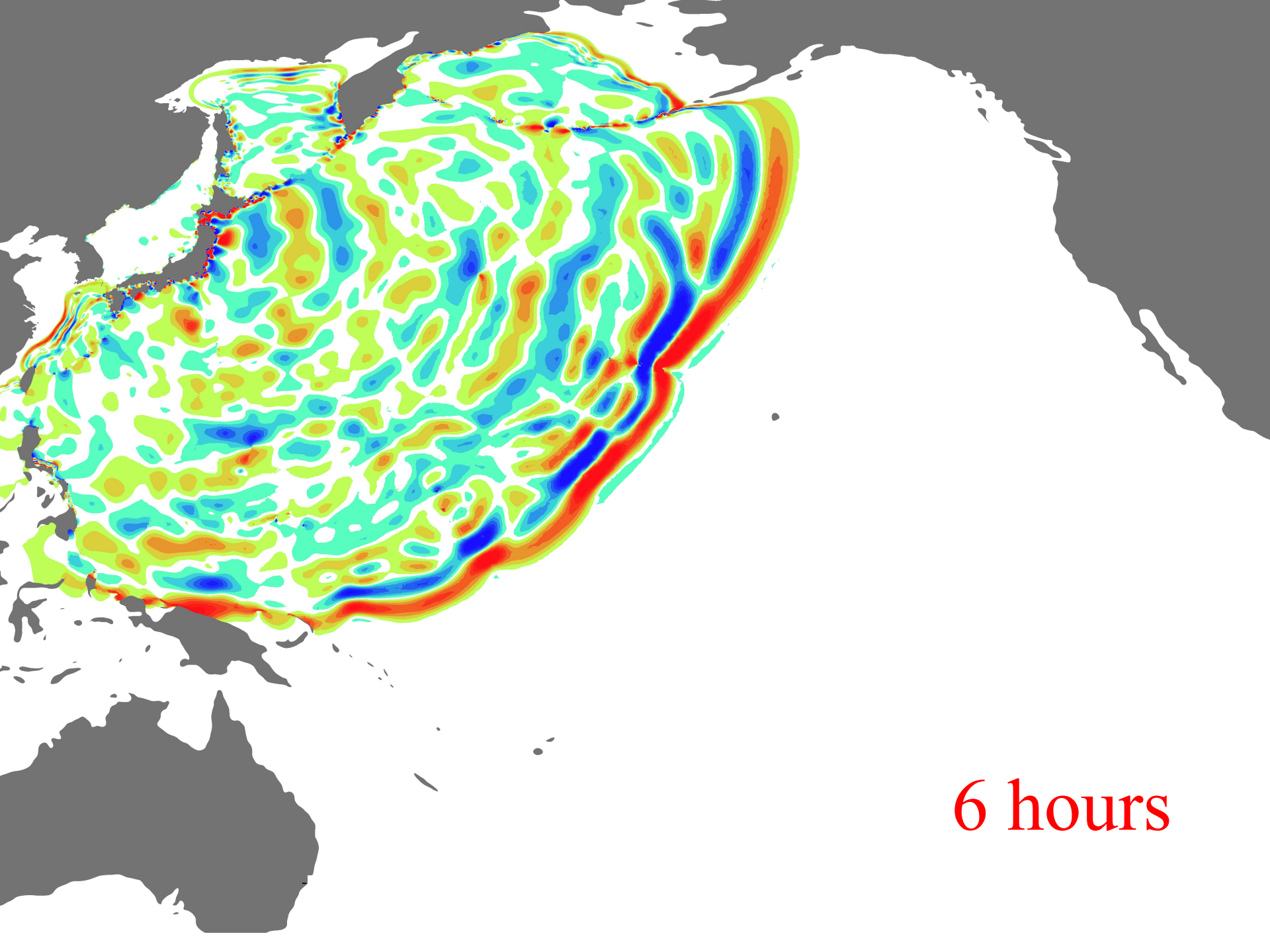
1 hour



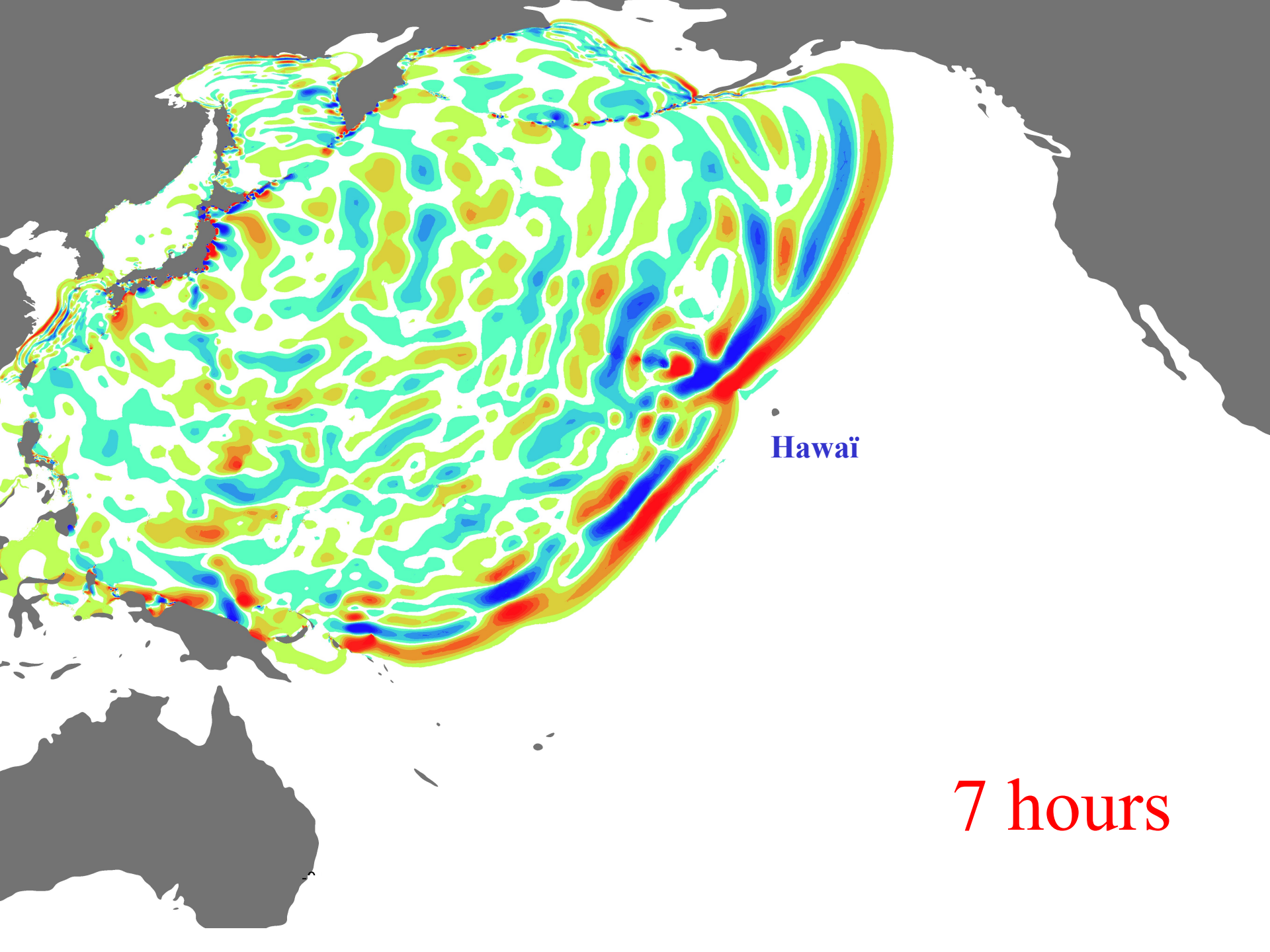
Indonesia

Hawai

4 hours

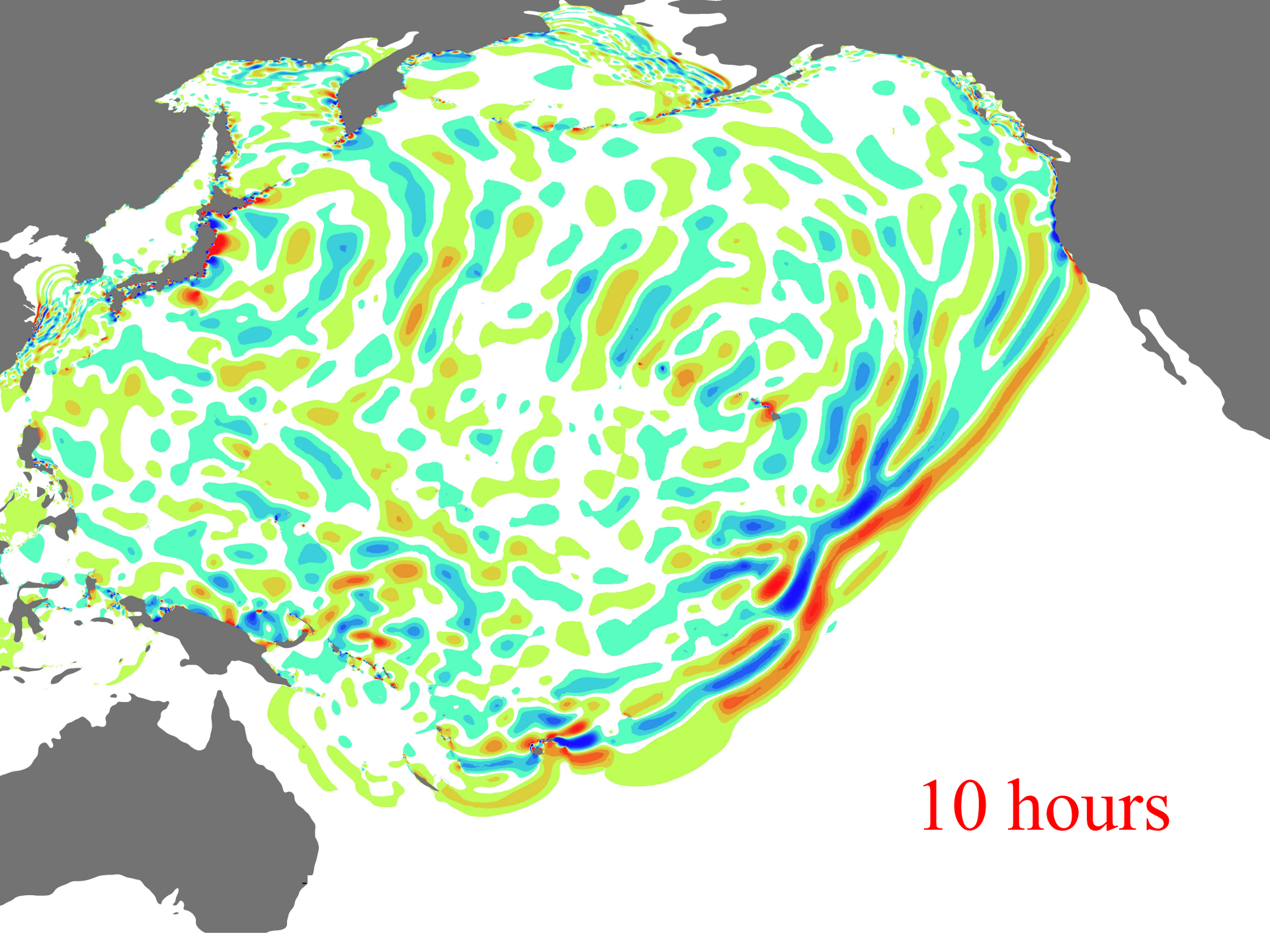


6 hours



Hawaiï

7 hours



10 hours

hello.c

```
#include <stdio.h>

int main()
{
    printf("hello, world\n");
    return 0;
}
```

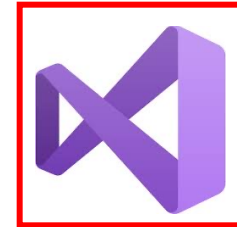
*Vous aimeriez apprendre à programmer,
mais vous ne savez pas par où commencer ?*

*(autrement dit : vous en avez marre des cours trop
compliqués que vous ne comprenez pas ? :-)*



```
Cours1 -- bash -- 71x16
Last login: Fri Feb 4 08:32:27 on ttys001
(base) mac-100-352:~ vl$ cd LaTeX/ep111110-2122/Slides/Cours1
(base) mac-100-352:Cours1 vl$ ls
a.out  hello.c
(base) mac-100-352:Cours1 vl$ touch hello.c
(base) mac-100-352:Cours1 vl$ cc hello.c
(base) mac-100-352:Cours1 vl$ ./a.out
hello, world
(base) mac-100-352:Cours1 vl$
```

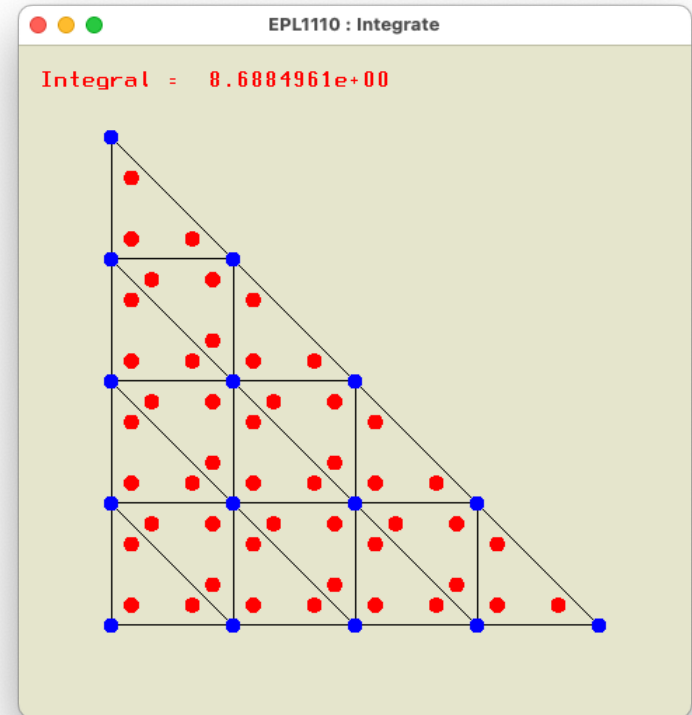
Comment compiler hello.c sur votre ordinateur ?



```
Cours1 -- -bash -- 93x27
(base) mac-100-352:Cours1 vl$
(base) mac-100-352:Cours1 vl$
(base) mac-100-352:Cours1 vl$
(base) mac-100-352:Cours1 vl$
(base) mac-100-352:Cours1 vl$
(base) mac-100-352:Cours1 vl$
(base) mac-100-352:Cours1 vl$
(base) mac-100-352:Cours1 vl$
(base) mac-100-352:Cours1 vl$
(base) mac-100-352:Cours1 vl$
(base) mac-100-352:Cours1 vl$
(base) mac-100-352:Cours1 vl$
(base) mac-100-352:Cours1 vl$
(base) mac-100-352:Cours1 vl$
(base) mac-100-352:Cours1 vl$
(base) mac-100-352:Cours1 vl$
(base) mac-100-352:Cours1 vl$ touch hello.c
(base) mac-100-352:Cours1 vl$ cc -c hello.c
(base) mac-100-352:Cours1 vl$ ls
hello.c hello.o
(base) mac-100-352:Cours1 vl$ cc -o hello hello.c
(base) mac-100-352:Cours1 vl$ ./hello
hello, world
(base) mac-100-352:Cours1 vl$ cc -o yep hello.o
(base) mac-100-352:Cours1 vl$ ./yep
hello, world
(base) mac-100-352:Cours1 vl$
```


Homework 1

$$\underbrace{\int_{\hat{\Omega}} f(x, y) \, dx \, dy}_I \approx \underbrace{\sum_{k=1}^3 w_k f(x_k, y_k)}_{I_h}$$



Ecrire la règle de Hammer

mainBasic.c

```
#include <stdio.h>
#include <math.h>

double integrate(double x[3], double y[3], double(*f)(double,double));
double integrateRecursive(double x[3], double y[3], double(*f)(double,double), int n);

double fun(double x, double y) { return cos(x) + y * y; }
double stupid(double x, double y) { return 1.0; }

int main()
{
    double x[3] = {0, 1, 0};
    double y[3] = {0, 0, 1};
    int n;
    printf("Surface integration      : %14.7e \n", integrate(x,y,stupid));
    printf("More funny integration : %14.7e \n", integrate(x,y,fun));
    for (n=0; n <= 4; n++) {
        double I = integrateRecursive(x,y,fun,n);
        printf("Recursive integration (n = %2d) : %14.7e \n", n,I);
    }
    return 0;
}
```

homework.c et mainBasic.c

```
double integrate(double x[3], double y[3], double (*f) (double, double))
{
    double I = 3.14;
    return I;
}
```

```
double integrateRecursive(double x[3], double y[3], double (*f)(double,double), int n)
{
    double I = 0.0;
    return I;
}
```

```
double fun(double x, double y) { return cos(x) + y * y; }
double stupid(double x, double y) { return 1.0; }
```

```
int main()
{
    double x[3] = {0, 1, 0};
    double y[3] = {0, 0, 1};
    int n;
    printf("Surface integration      : %14.7e \n", integrate(x,y,stupid));
    printf("More funny integration : %14.7e \n", integrate(x,y,fun));
    for (n=0; n <= 1; n++) {
        double I = integrateRecursive(x,y,fun,n);
        printf("Recursive integration (n = %2d) : %14.7e \n", n,I); }
    return 0;
}
```

Comment compiler le devoir sur votre ordinateur ?



homework.c



homeworkSoluce.c



mainBasic.c

```
Cours1 — -bash — 79x29
(base) mac-1U0-352:Cours1 vl$
(base) mac-1U0-352:Cours1 vl$
(base) mac-1U0-352:Cours1 vl$
(base) mac-1U0-352:Cours1 vl$
(base) mac-1U0-352:Cours1 vl$
(base) mac-1U0-352:Cours1 vl$
(base) mac-1U0-352:Cours1 vl$
(base) mac-1U0-352:Cours1 vl$
(base) mac-1U0-352:Cours1 vl$
(base) mac-1U0-352:Cours1 vl$
(base) mac-1U0-352:Cours1 vl$ cc -o myFem mainBasic.c homework.c
(base) mac-1U0-352:Cours1 vl$ ./myFem
Surface integration      : 3.1400000e+00
More funny integration  : 3.1400000e+00
Recursive integration (n = 0) : 0.0000000e+00
Recursive integration (n = 1) : 0.0000000e+00
Recursive integration (n = 2) : 0.0000000e+00
Recursive integration (n = 3) : 0.0000000e+00
Recursive integration (n = 4) : 0.0000000e+00
(base) mac-1U0-352:Cours1 vl$ cc -o myFem mainBasic.c homeworkSoluce.c
(base) mac-1U0-352:Cours1 vl$ ./myFem
Surface integration      : 5.0000000e-01
More funny integration  : 5.4302895e-01
Recursive integration (n = 0) : 5.4302895e-01
Recursive integration (n = 1) : 2.1721229e+00
Recursive integration (n = 2) : 8.6884961e+00
Recursive integration (n = 3) : 3.4753986e+01
Recursive integration (n = 4) : 1.3901594e+02
(base) mac-1U0-352:Cours1 vl$
```


Bon : c'est aussi simple ?

perso.uclouvain.be/vincent.l

Yahoo Apple Twitter Gazou Wikipedia Informations Divers Php Actions Cours kine Altanta

linux logo - Recherche Google

EPL1110 News Horaire Documents

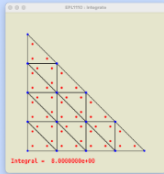
 Introduction aux éléments finis (LEPL1110)
Vincent Legat
Jean-François Remacle
Louvain School of Engineering
Université catholique de Louvain

Il faut d'abord t'identifier :-)

News Documents Videos & podcasts !

Comment obtenir un exécutable sur votre ordinateur ? Programmes Python Devoirs en C

Devoirs en C...
Les énoncés seront disponibles progressivement...



Devoir 1 : Integrate (02-02-2022)
Projet à télécharger : Integrate.zip
Enoncé du devoir : Integrate.pdf
Deadline : **Lundi 14 février 2022 à 23h59**

© 2020 Vincent Legat Contact - Support

```
src -- -bash -- 90x27
IntegrateWithoutBov      homework.c      myFem
hello                   homeworkSoluce.c  yep
hello.c                 mainBasic.c
(base) mac-1U0-352:Cours1 v1$ cd Integrate
(base) mac-1U0-352:Integrate v1$ ls
CMakeLists.txt  glfw      src
(base) mac-1U0-352:Integrate v1$ cd src
(base) mac-1U0-352:src v1$ ls
fem.c           glfem.c      homework.c
fem.h           glfem.h      main.c
(base) mac-1U0-352:src v1$ gcc -o myFem *.c
In file included from glfem.c:10:
./glfem.h:21:10: fatal error: 'GLFW/glfw3.h' file not found
#include <GLFW/glfw3.h>
               ^~~~~~
1 error generated.
In file included from homework.c:3:
./glfem.h:21:10: fatal error: 'GLFW/glfw3.h' file not found
#include <GLFW/glfw3.h>
               ^~~~~~
1 error generated.
In file included from main.c:5:
./glfem.h:21:10: fatal error: 'GLFW/glfw3.h' file not found
#include <GLFW/glfw3.h>
               ^~~~~~
1 error generated.
(base) mac-1U0-352:src v1$
```

Eh non !

perso.uclouvain.be/vincent.legat/zouLab/ep1110.php?action=

Yahoo Apple Twitter Gazou Wikipedia Informations Divers Php Actions Cours kine Altanta Finite Elements Méthodes numériques

linux logo - Recherche Google

EPL1110 News Horaire Documents Mon profil Mon binome

Exécution et soumission d'un programme sur le serveur...

Groupe : 1 (viegat-jeremacle)
Binôme : Remacle, Jean-François
Deadline : February 14 2022 23:59:59.
Now : February 04 2022 09:04:25.

```
1 #include <math.h>
2
3
4 #include "g1fem.h"
5
6
7
8
9
10
11
```

Position: Ln 11, Ch 1 Total: Ln 65, Ch 1656

Exécuter le programme sur le serveur Voir le diagnostic

Valider son programme

© 2020 Vincent Legat

perso.uclouvain.be/vincent.legat/zouLab/ep1110.php?...

Yahoo Apple Twitter Gazou Wikipedia Informations Divers Php Actions Cours kine Altanta Finite Elements Méthodes numériques

linux logo - Recherche Google

EPL1110 News Horaire Documents Mon profil Mon binome Hello Vincent Deconnexion

Introduction aux éléments finis (LEPL1110)

Vincent Legat
Jean-François Remacle
Louvain School of Engineering
Université catholique de Louvain

News Documents Videos & podcasts !

Comment obtenir un exécutable sur votre ordinateur ? Programmes Python Devoirs en C

Liste des étudiants Liste des groupes Equipe didactique Former son groupe

Soumettre le devoir 1 *Integrate*

© 2020 Vincent Legat Contact - Support

Soumettre votre devoir !

**Tous les devoirs seront corrigés de manière automatique !
Bien veiller à ce que la version soumise soit bien compilée !
Aucune soumission tardive ne sera admise !
Les devoirs sont réalisés individuellement !**

perso.uclouvain.be/vincent.legat/zouLab/epi

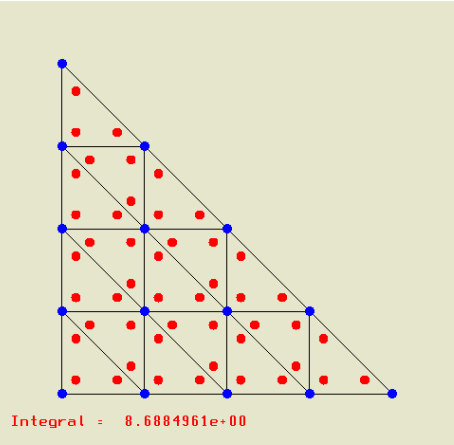
Yahoo Apple Twitter Gazou Wikipedia Informations Divers Php Actions Cours kine Altanta Finite Elements

linux logo - Recherche Google

EPL1110 News Horaire Documents Mon profil Mon binome

Exécution et soumission d'un programme sur le serveur...

Groupe : 1 (vlegat-jeremacle)
Binôme : Remacle, Jean-François
Deadline : February 14 2022 23:59:59.
Now : February 04 2022 09:06:11.



Integral = 8.6884961e+00

© 2020 Vincent Legat

perso.uclouvain.be/vincent.legat/zouLab/epi

Yahoo Apple Twitter Gazou Wikipedia Informations Divers Php Actions Cours kine Altanta Finite Elements

linux logo - Recherche Google

EPL1110 News Horaire Documents Mon profil Mon binome Hello Vincent Deconnexion

```
int i,j;
const int nodes[4][3] = {{0,3,5},{3,1,4},{5,4,2},{3,4,5}};
const double xsi[6] = {0.0,1.0,0.0,0.5,0.5,0.0};
const double eta[6] = {0.0,0.0,1.0,0.0,0.5,0.5};
double xLoc[3];
double yLoc[3];

if (n <= 0) return integrate(x,y,f);

double I = 0.0;
for (i=0; i<4; i++) {
  for (j=0; j<3; j++) {
    double xsiLoc = xsi[nodes[i][j]];
    double etaLoc = eta[nodes[i][j]];
    xLoc[j] = interpolate(x,xsiLoc,etaLoc);
    yLoc[j] = interpolate(y,xsiLoc,etaLoc); }
  I += integrateRecursive(xLoc,yLoc,f,n-1); }

return I;
}
```

👍 👍 👍 👍 👍 ❤️ ❤️ ❤️ ❤️

Faire une nouvelle soumission Voir le diagnostic

Valider son programme

© 2020 Vincent Legat Contact - Support

Valider
et verifier son devoir !

Le C est un langage de bas niveau : les pointeurs ;-(

```
int main(void)
{
    int a = 4;
    printf(" ====   a === %d \n", a);
    printf(" ====   &a === %d \n", &a);
    int *b = &a; // &&a do not exist : why ?
    printf(" ====   &&a === %d \n", &b);
    exit(0);
}
```

Adresse de la case mémoire

int	a	1606415436	4
*int	&a	1606415424	1606415436
**int	&&a	...	1606415424

Valeur

L'utilisation des pointeurs permet d'écrire des codes très efficaces et rapides...
Par contre, programmer est une tâche plus délicate et fastidieuse.
Mais, la rapidité d'un code est critique pour la simulation numérique (et le jeux !) :
Le langage C (ou C++) est bon choix ici !

Le C est un langage de bas niveau les pointeurs ;-(

```
int a = 0;
int *b = &a;
b[0] = 4;
printf("==== *b === %d \n", *b);
printf("==== b[0] === %d \n", b[0]);
printf("==== a === %d \n", a);
printf("==== b === %d \n", b);
```

```
int    a    *b    b[0]
*int   &a    b
```

Adresse de la
case mémoire

1606415436	4
1606415424	1606415436

**L'utilisation des pointeurs permet d'écrire des codes très efficaces et rapides...
Par contre, programmer est une tâche plus délicate et fastidieuse.
Mais, la rapidité d'un code est critique pour la simulation numérique (et le jeu !) :**
Le langage C (ou C++) est bon choix ici !

On peut écrire n'importe où dans
la mémoire de l'ordinateur !
Ouuuuuups : c'est pas joli

```
int a = 0;
int *b = &a;
b[0] = 4;
b[1] = 3;
printf("==== *b === %d \n", *b);
printf("==== b[0] === %d \n", b[0]);
printf("==== b[1] === %d \n", b[1]);
printf("==== a === %d \n", a);
printf("==== b === %d \n", b);
printf("==== &b[0] == %d \n", &b[0]);
printf("==== &b[1] == %d \n", &b[1]);
```

Et le pire, c'est que cela marche parfois...
Parfois pas : **Segmentation fault**

int		b[1]	1606415440	3
int	a	*b	1606415436	4
*int	&a	b	1606415424	1606415436

Adresse de la
case mémoire

homework.c

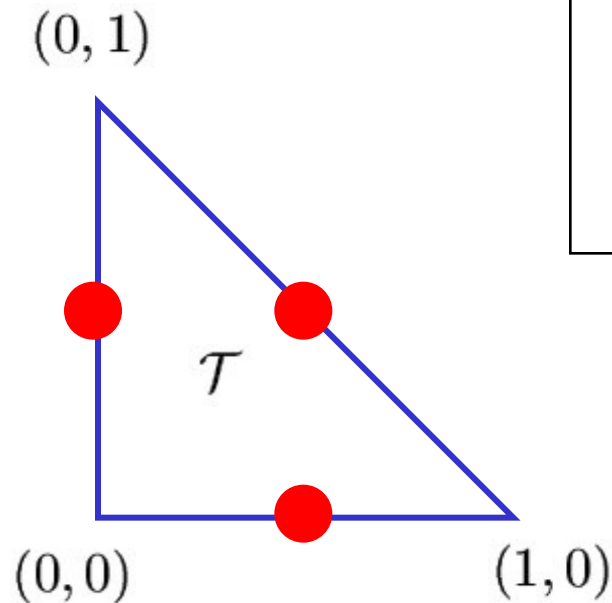
La solution doit se trouver dans le fichier homework.c
uniquement.... On ne regarde jamais main.c !

```
#include <stdio.h>
#include <math.h>

double integrate(double x[3], double y[3], double (*f) (double, double))
{
    double I = 3.14;
    int I;
    for (i=0; i<3; i++)
        printf("    == node %d : %14.7e %14.7e \n",i+1,x[i],y[i]);
    return I;
}

double integrateRecursive(double x[3], double y[3], double (*f)(double,double), int n)
{
    double I = 0.0;
    return I;
}
```


Intégration sur un triangle : Règle de Hammer à 3 points



$$\underbrace{\int_{\mathcal{T}} f(x, y) \, dx \, dy}_{I} \approx \underbrace{\sum_{k=1}^3 w_k f(X_k, Y_k)}_{I^h}$$

	X_k	Y_k	w_k
1	0.5	0.0	1/6
2	0.5	0.5	1/6
3	0.0	0.5	1/6

Démontrer que la formule de Hammer à trois points permet d'intégrer exactement n'importe quel polynôme à deux variables de degré deux : $a + bx + cy + dx^2 + ey^2 + fxy$

Question

$$\begin{aligned}
 I &= \frac{a}{2} + b \int_0^1 x \int_0^{1-x} dy \, dx + c \int_0^1 y \int_0^{1-y} dx \, dy \\
 &\quad + d \int_0^1 x^2 \int_0^{1-x} dy \, dx + e \int_0^1 y^2 \int_0^{1-y} dx \, dy + f \int_0^1 x \int_0^{1-x} y \, dy \, dx \\
 &= \frac{a}{2} + b \left[\frac{x^2}{2} - \frac{x^3}{3} \right]_0^1 + c \left[\frac{y^2}{2} - \frac{y^3}{3} \right]_0^1 \\
 &\quad + d \left[\frac{x^3}{3} - \frac{x^4}{4} \right]_0^1 + e \left[\frac{y^3}{3} - \frac{y^4}{4} \right]_0^1 + f \left[\frac{x^2}{4} - \frac{x^3}{3} + \frac{x^4}{8} \right]_0^1 \\
 &= \frac{a}{2} + \frac{b}{6} + \frac{c}{6} + \frac{d}{12} + \frac{e}{12} + \frac{f}{24} \\
 &= I^h
 \end{aligned}$$



**Degré de
précision**

Et un autre triangle ?

$$\begin{aligned}x' &= 10x \\ y' &= 15 - 15y\end{aligned}$$

