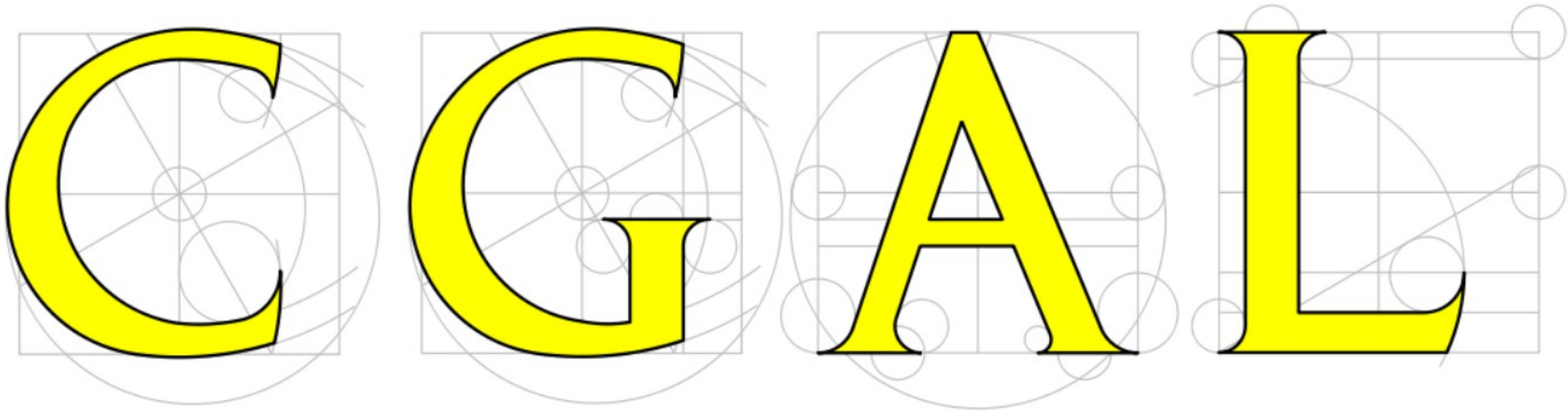
La Bibliothèque



Computational Geometry Algorithms Library

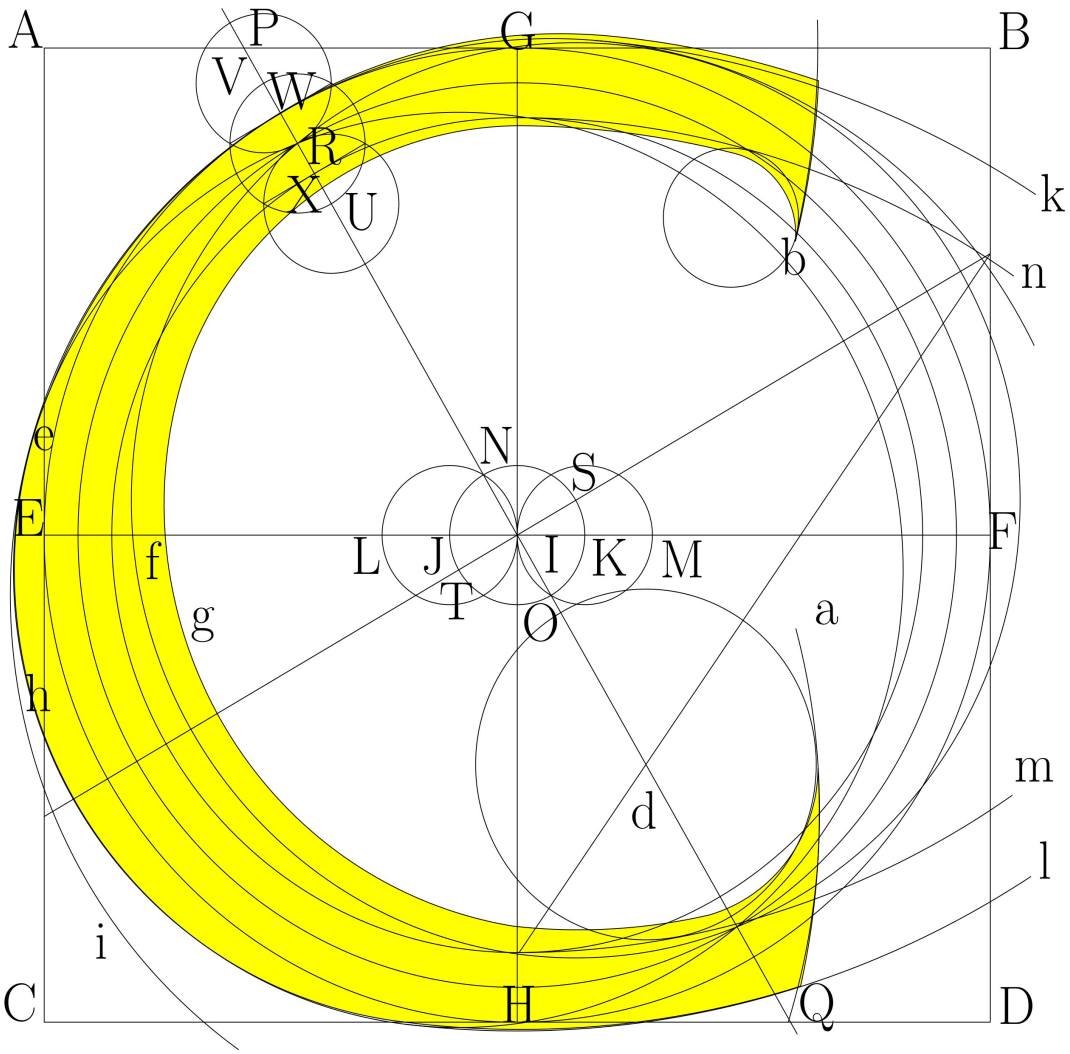
Sylvain Pion

Collège de France

26 Avril 2017

Plan

- Projet CGAL
- Contenu de la bibliothèque
- Généricité
- Robustesse numérique



Le Projet CGAL

Mission Originale

"Rendre l'ensemble des algorithmes géométriques développés dans le domaine de la géométrie algorithmique disponibles pour les applications industrielles"

Proposition de Projet Européen CGAL, **1996**

Organisation du projet

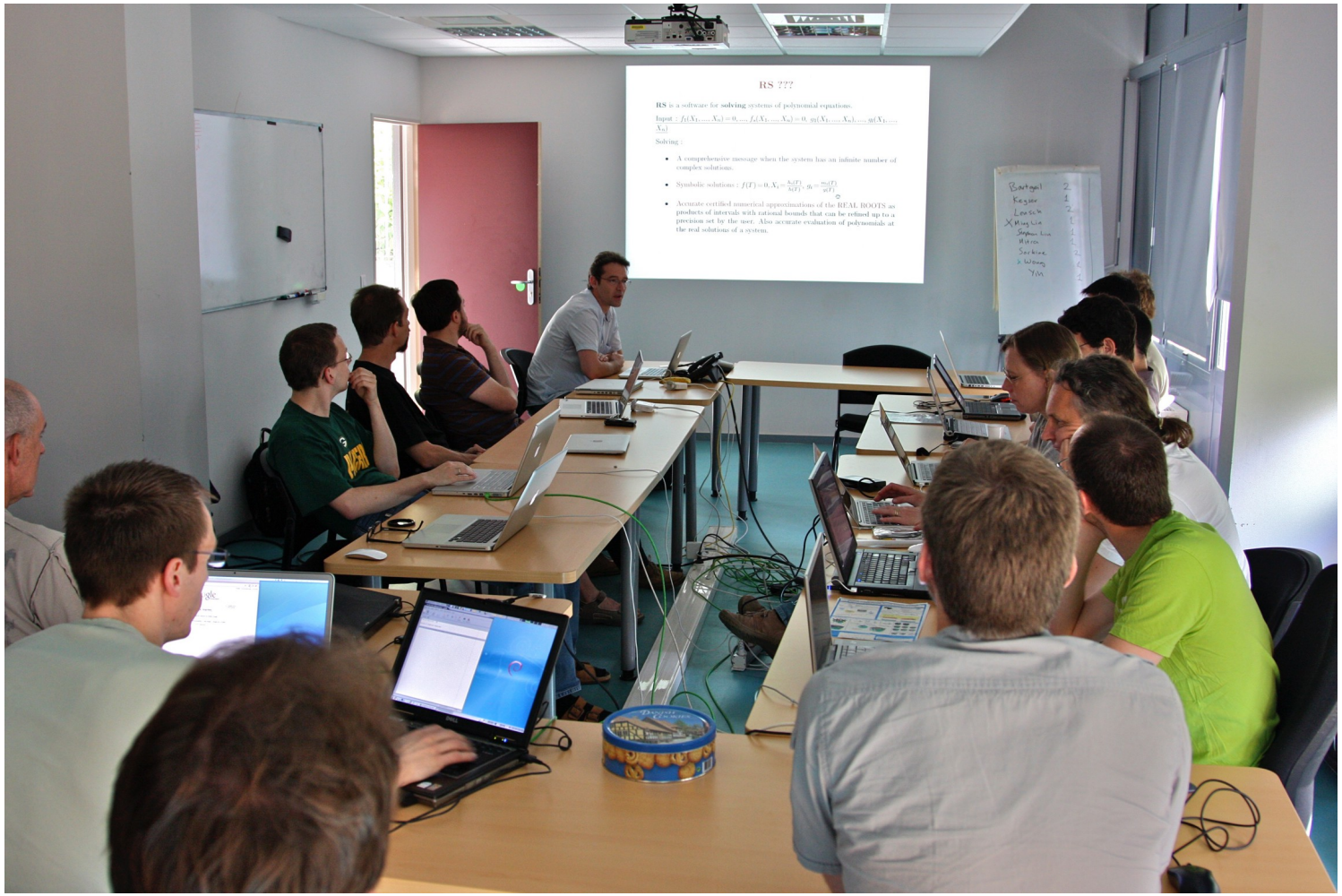
- Partenaires académiques :

  INRIA, Max-Planck Institute, U Tel-Aviv, ETH Zurich...

- Comité éditorial (CGAL Editorial Board)
  - Pilote et anime le projet
  - Relecture des propositions

- Infrastructure de développement
  - GitHub, tests journaliers (~30 configurations)
  - Réunions biannuelles des développeurs

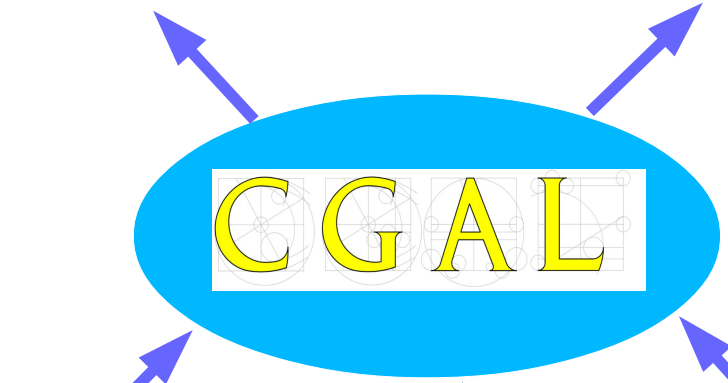
Réunion de développeurs



CGAL en quelques chiffres

1,100,000	lignes de code C++
10,000	téléchargements/an
5,000	pages de manuel
4,000	inscrits à cgal-announce@
2,000	inscrits à cgal-discuss@
200	utilisateurs commerciaux
130	composants logiciels
100	contributeurs individuels
6	mois entre chaque version
2	licences: GPL + commerciale
?	publications scientifiques

Eco-système



Utilisateurs Open Source

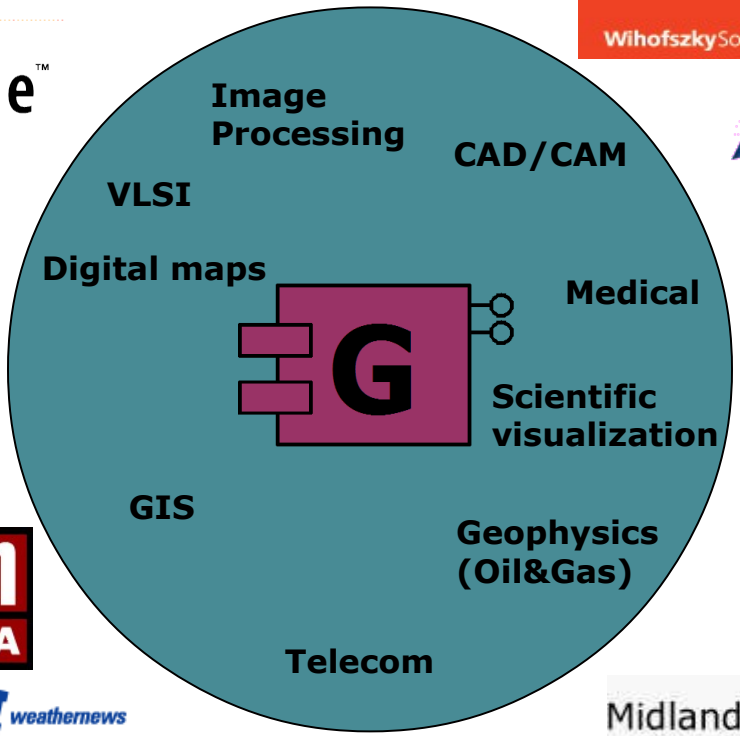
Utilisateurs Commerciaux

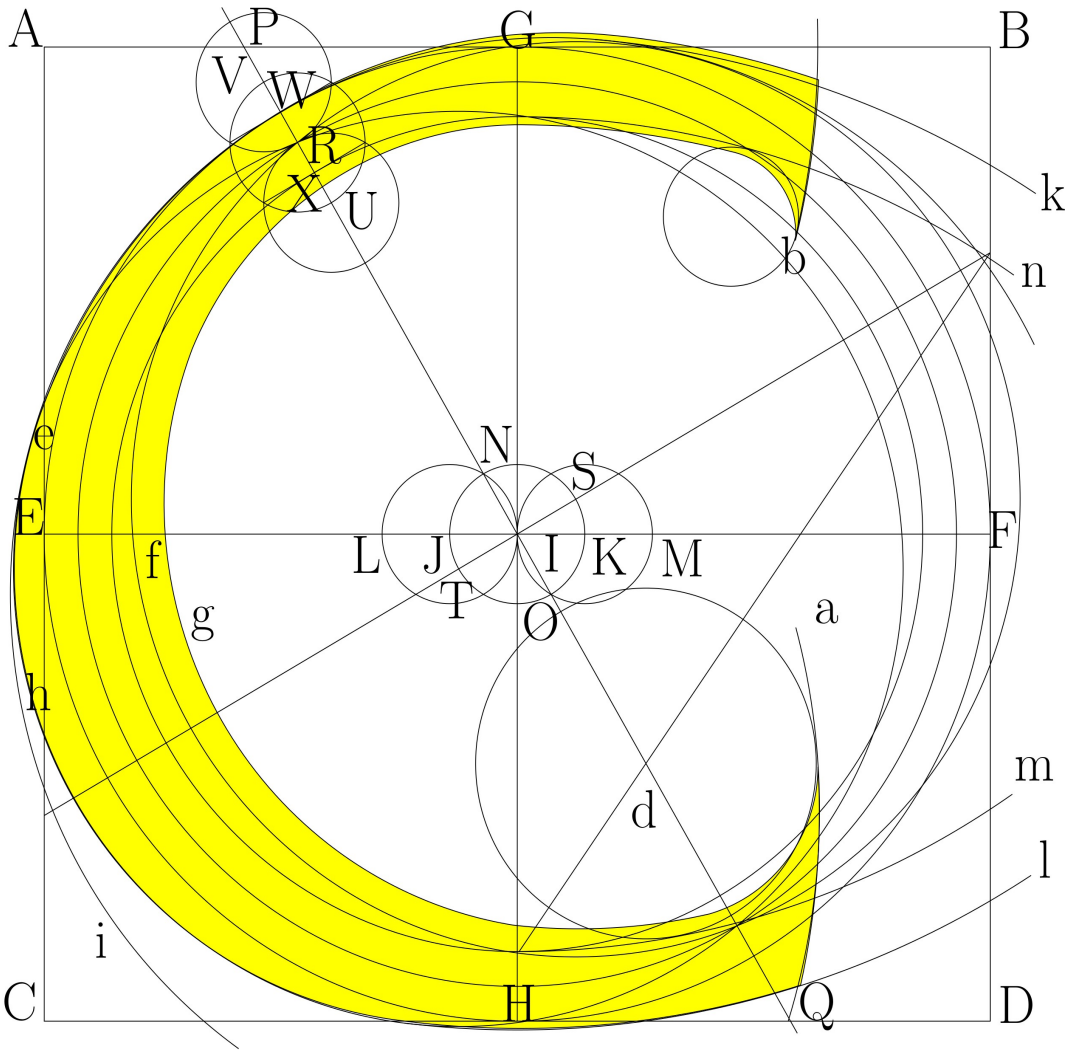
Contributeurs Open Source

Chercheurs

Geometry Factory

Utilisateurs Commerciaux

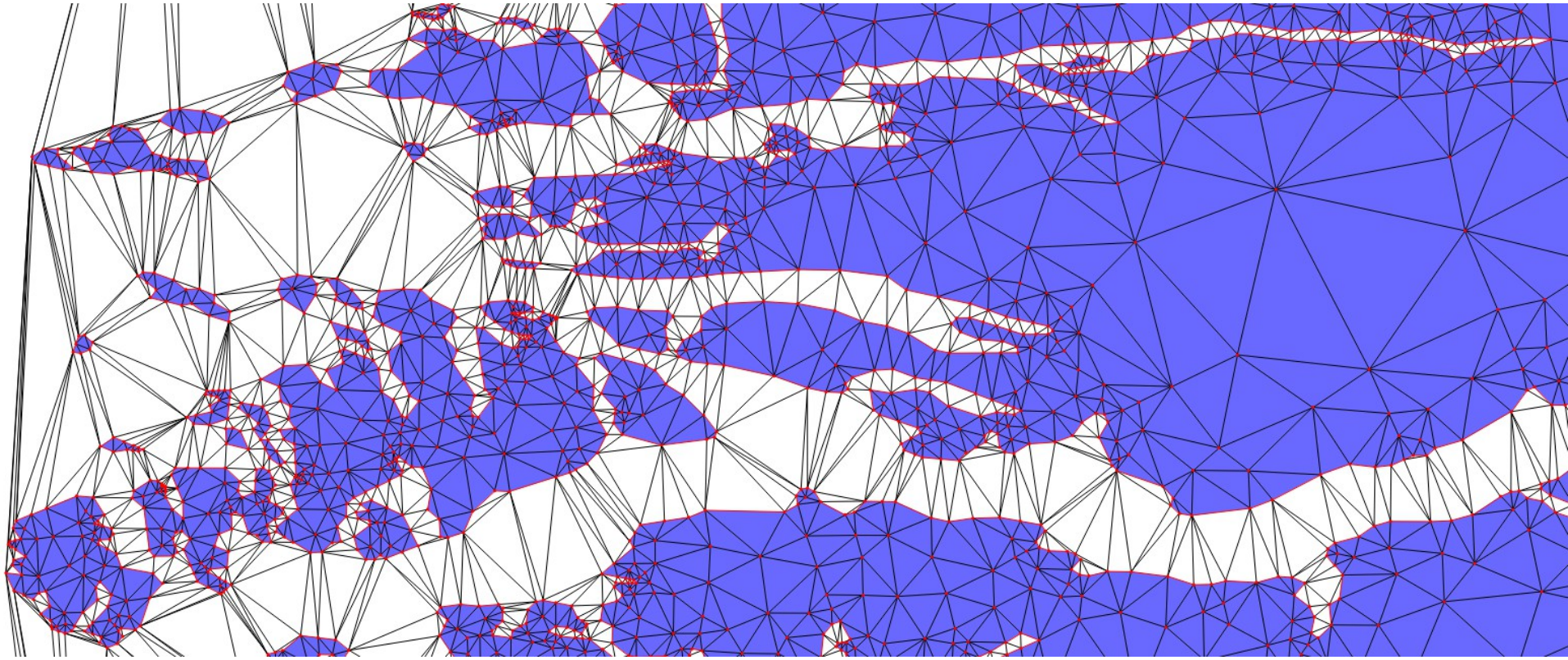




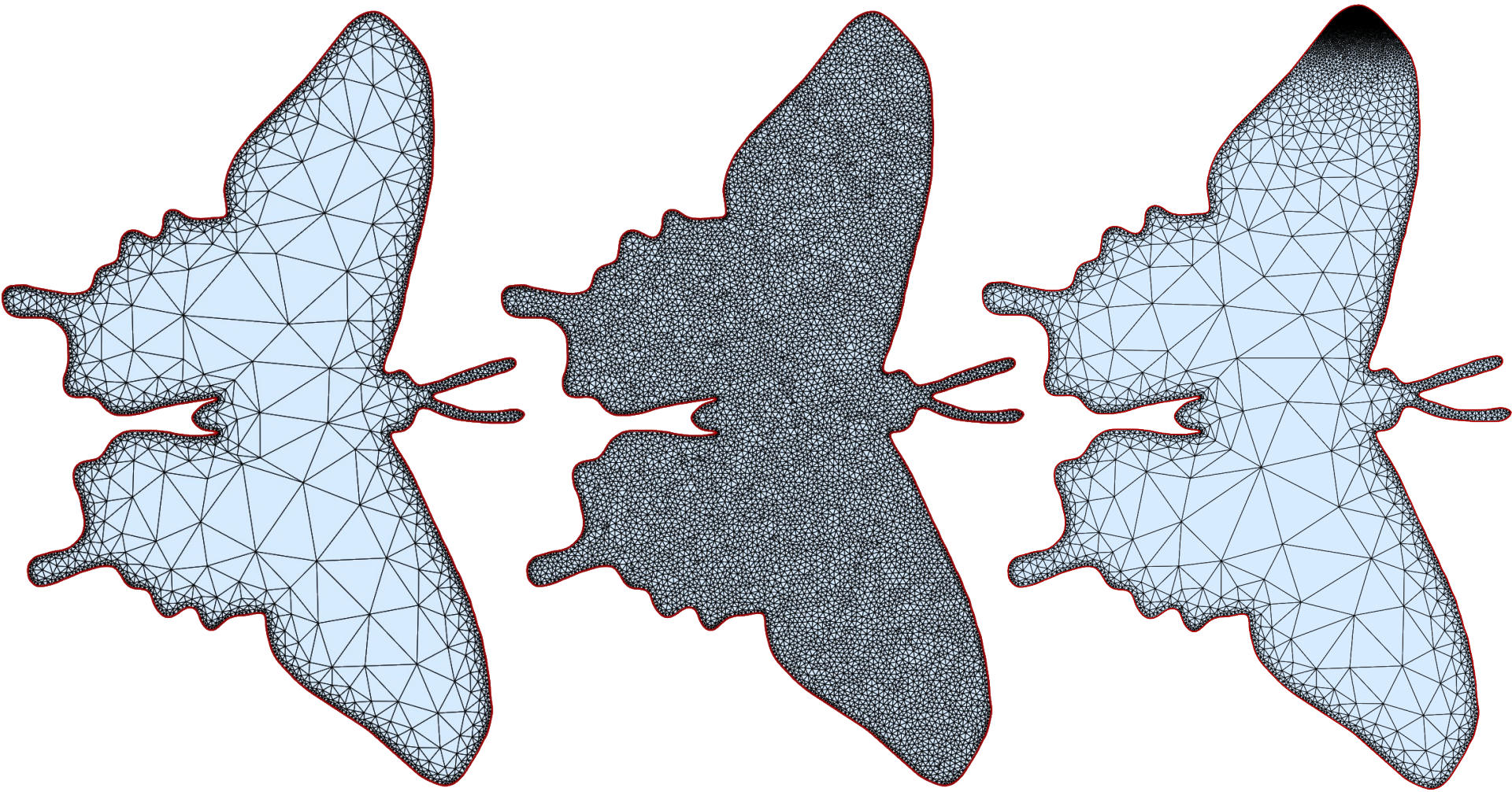
Contenu de CGAL

Triangulations de Polygones

Triangulations de Polygones



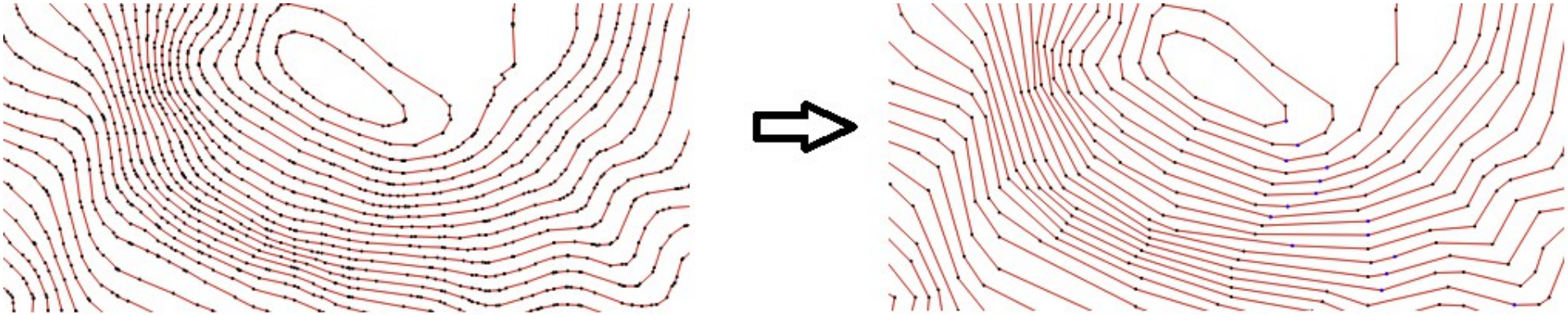
Maillages paramétrables



Sans contrainte | Uniforme | Fonction de taille

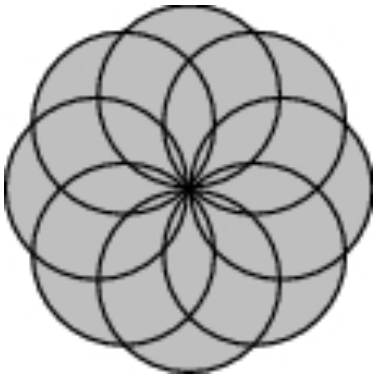
Simplification Simultanée de Polylignes

- Garantit qu'après simplification
  - les îles restent des îles
  - les isolignes ne se coupent pas

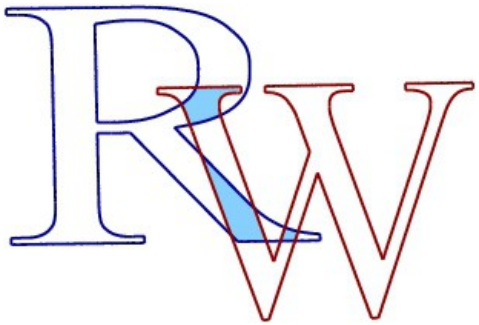


Opérations Booléennes

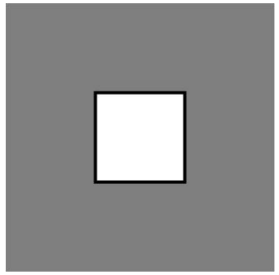
Union



Intersection



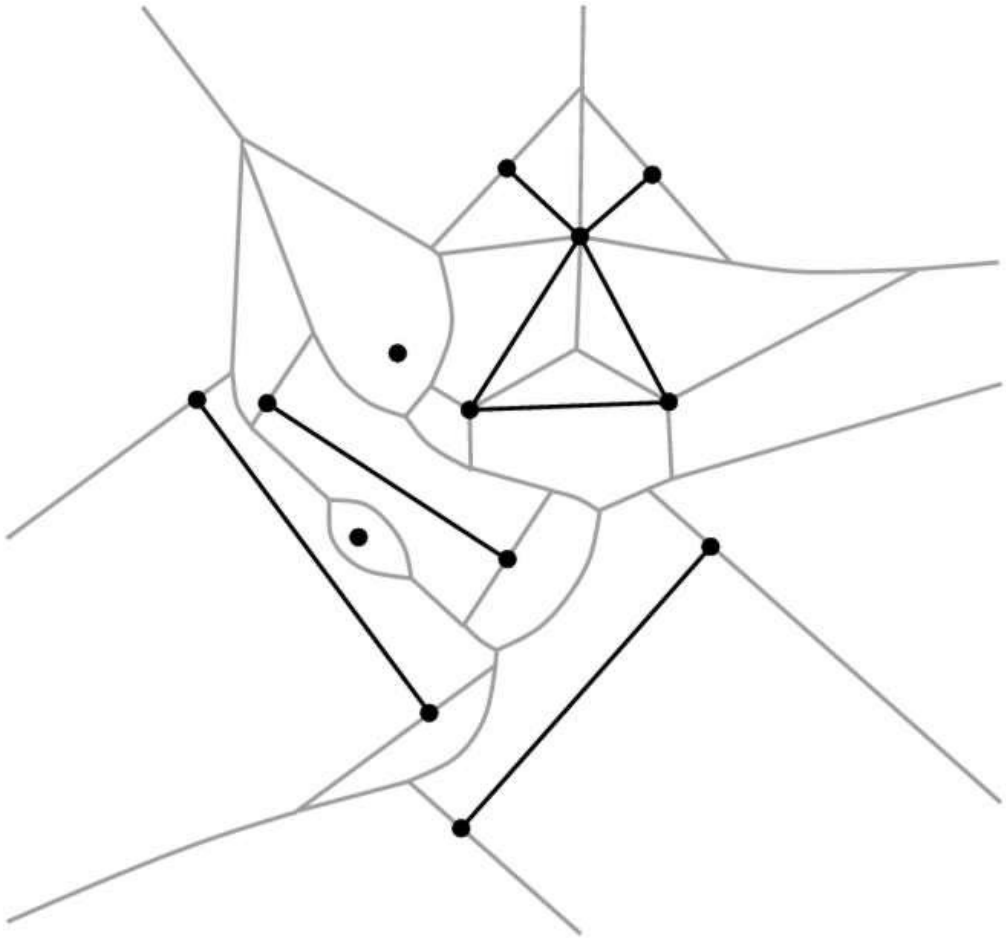
Complément



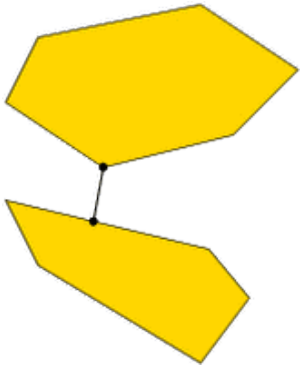
Supporte explicitement :

Arcs de Cercles, Courbes de Bézier, Segments de droites

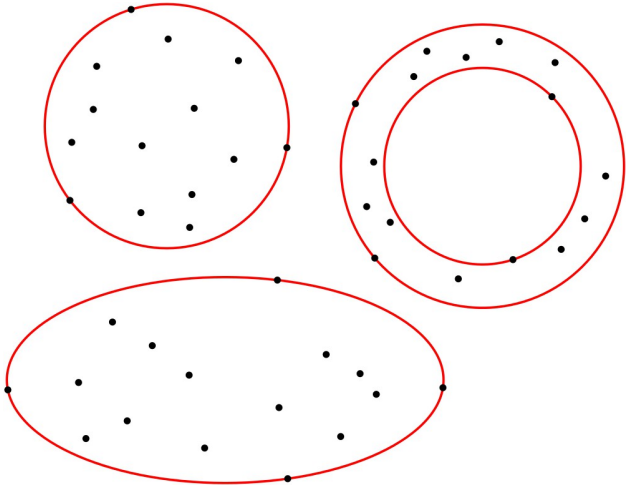
Diagrammes de Voronoi de Segments



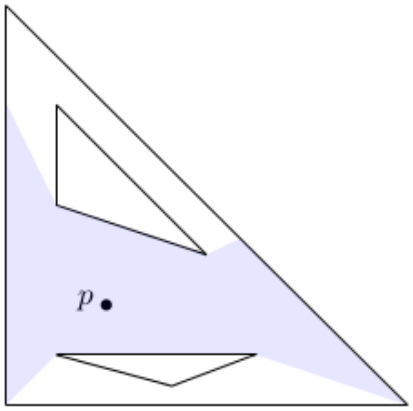
Autres Algorithmes



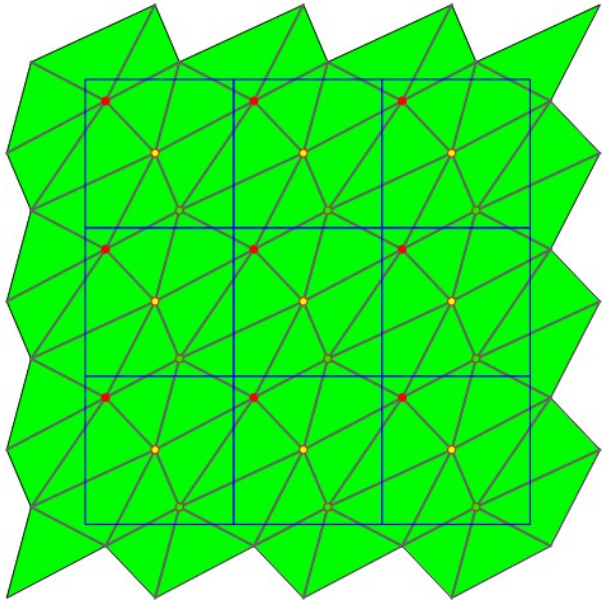
Distance Optimale



Volume englobant



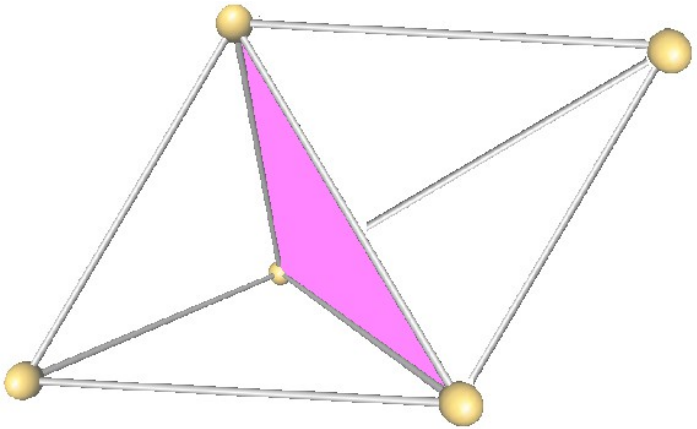
Polygone de Visibilité

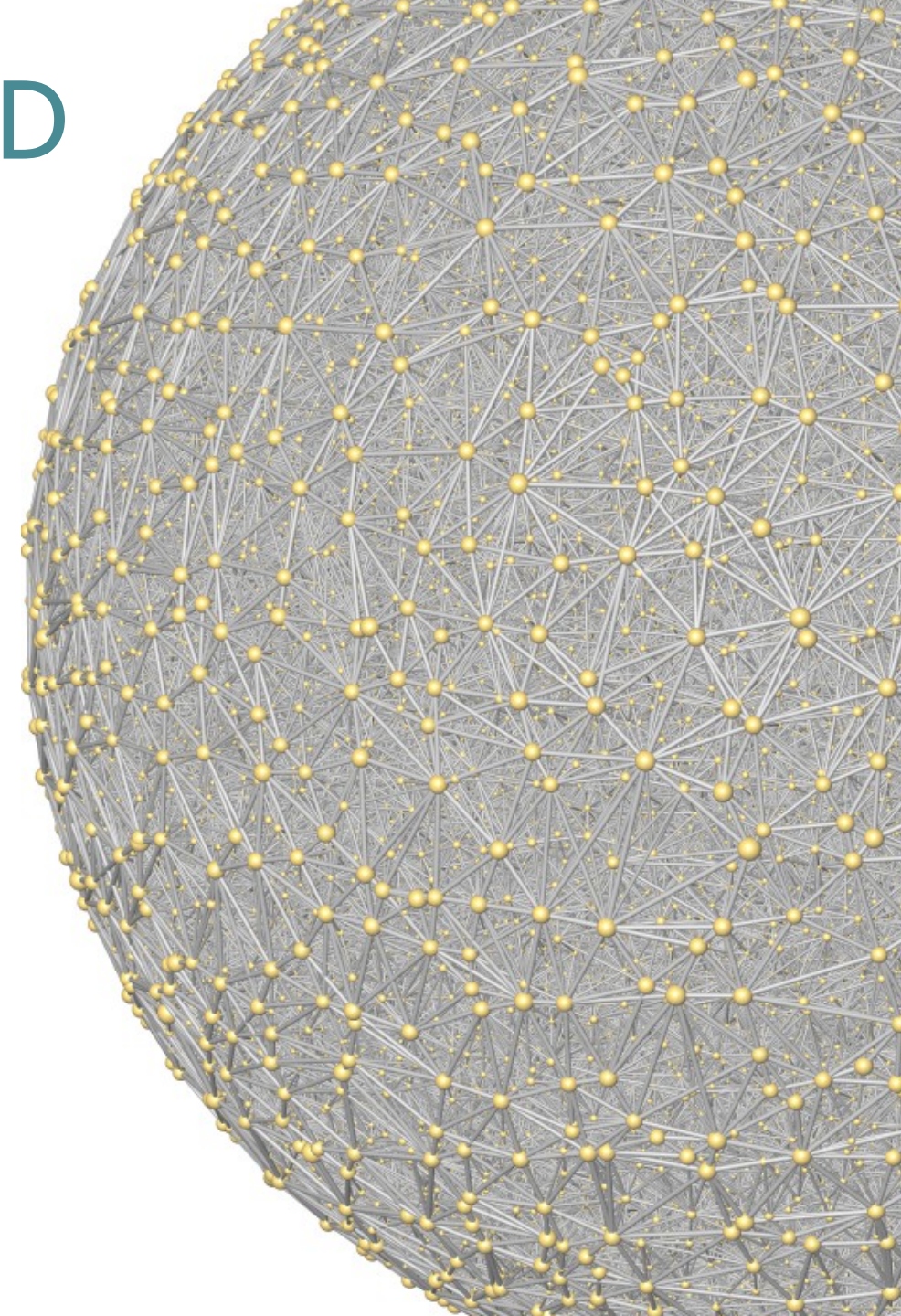


Triangulation de Delaunay périodique

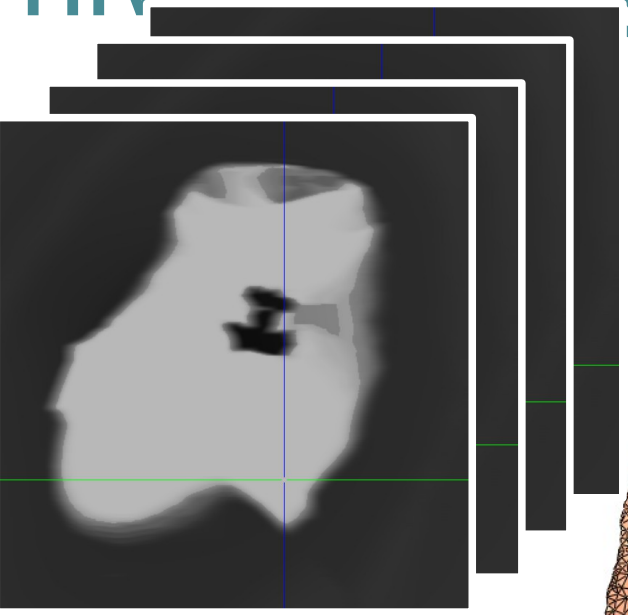
Triangulations 3D

- Delaunay et régulière
- Entièrement dynamique
- 1M points 3D en 16s



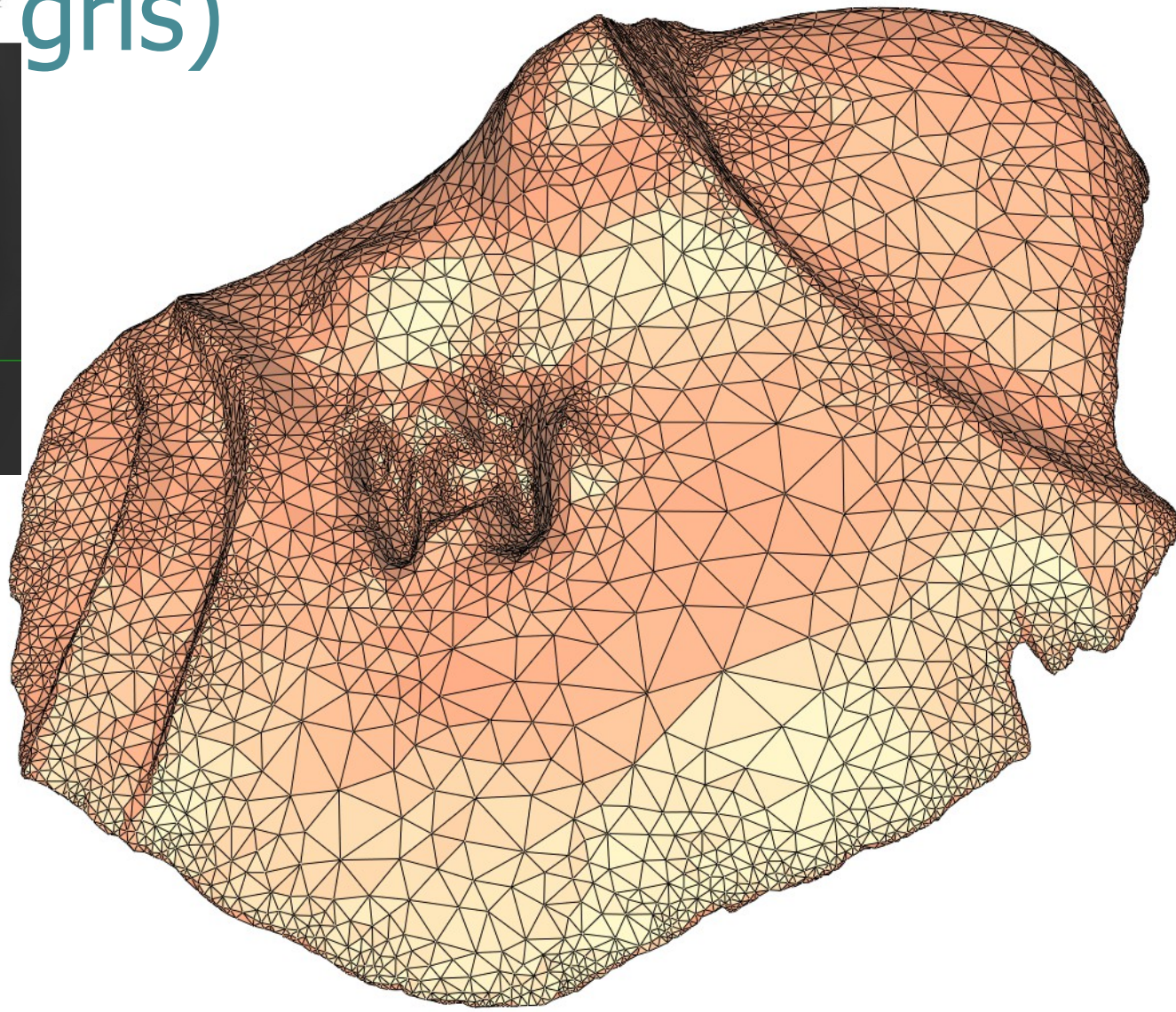


Maillage Surfacique (Image à niveaux de gris)

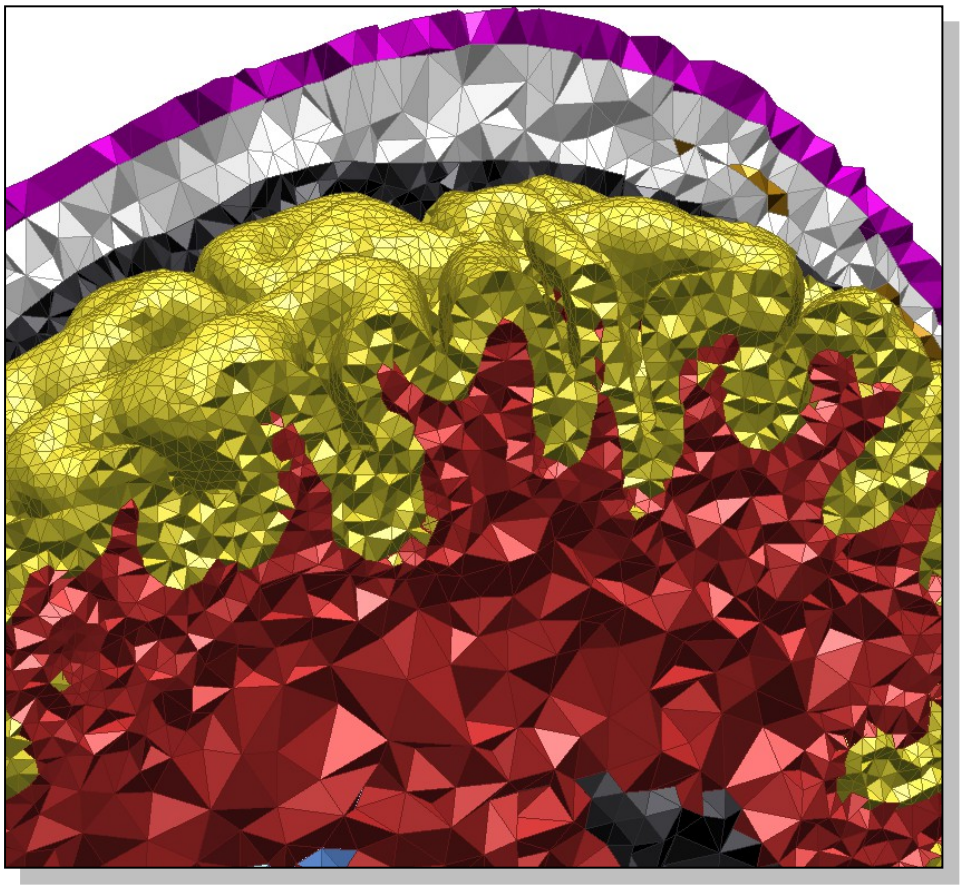


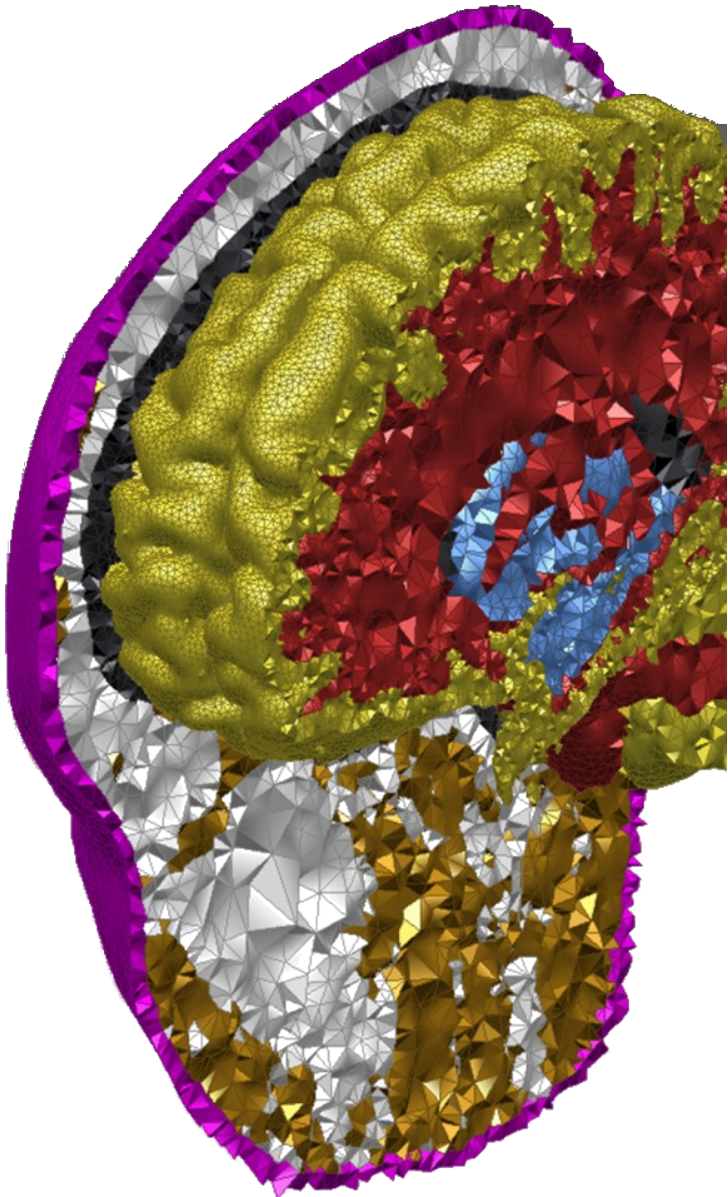
Input:
3D voxel data for SEG Salt Model

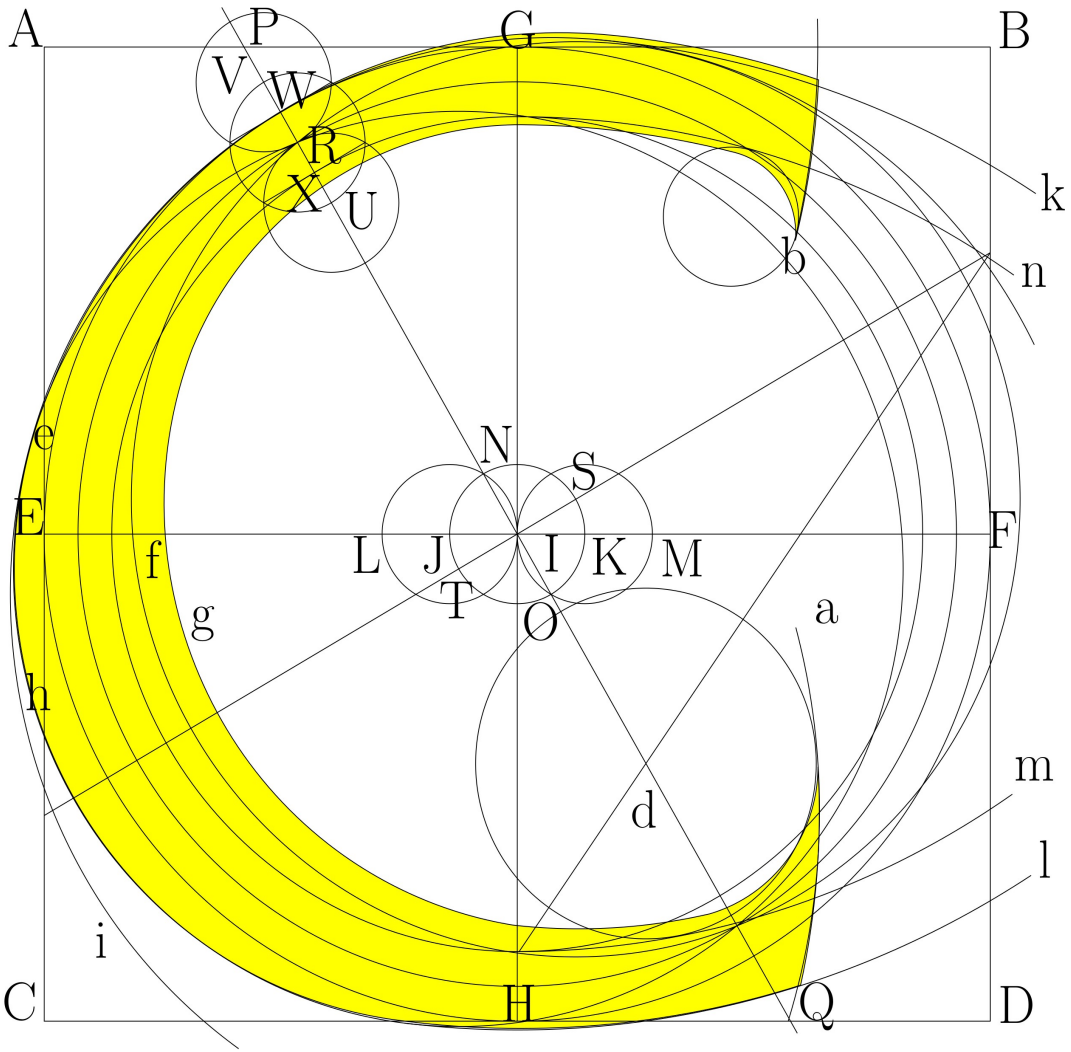
Triangle surface mesh approximating isosurface of input 3D image



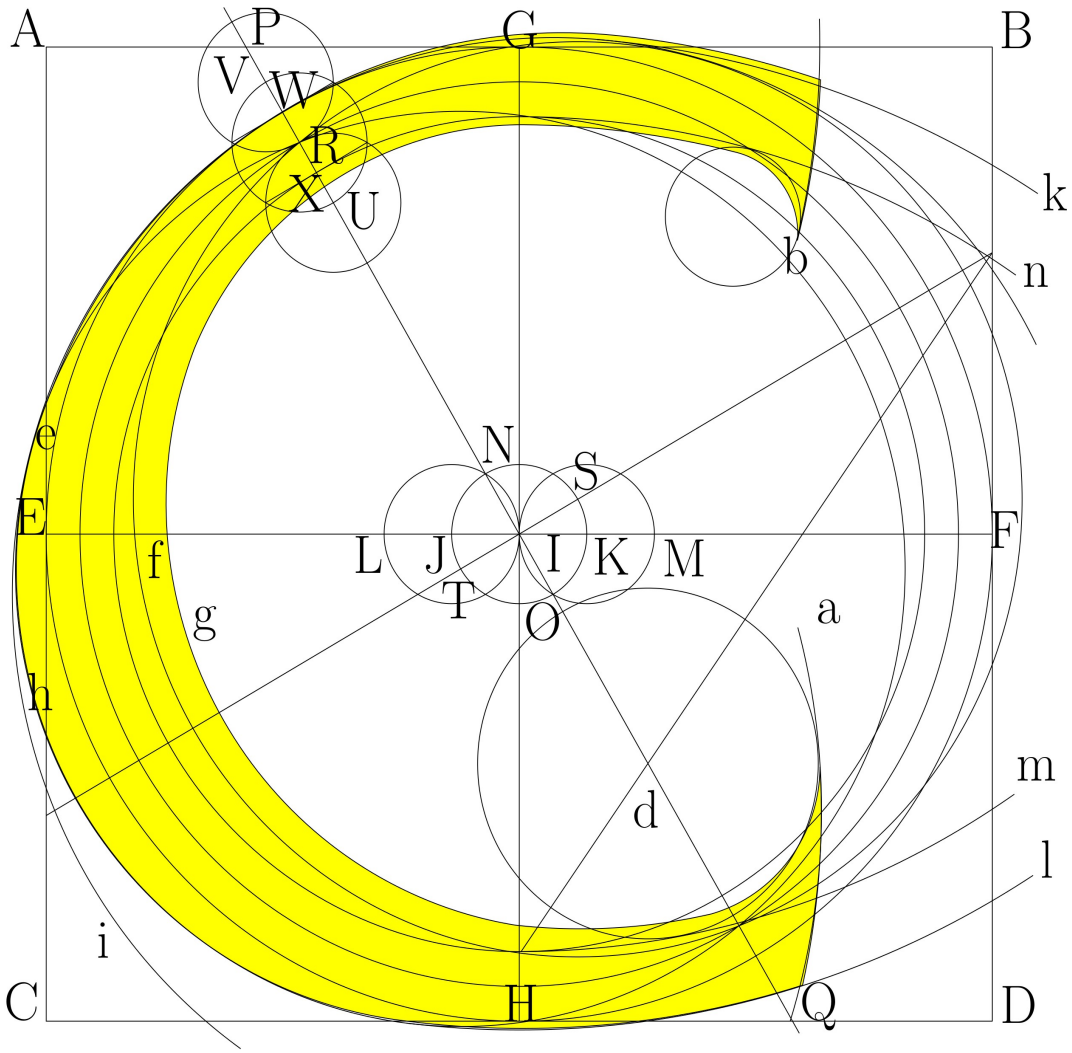
Maillage Volumique (Image segmentée)







Fondations : Généricité et Robustesse



Programmation Générique

Généricité de la STL

```
template <class Key, class Less>
class set {
  Less less;

  insert(Key k)
  {
     if (less(k, treenode.key)(
       insertLeft(k);
     else
       insertRight(k);
  }
};
```

Généricité dans CGAL

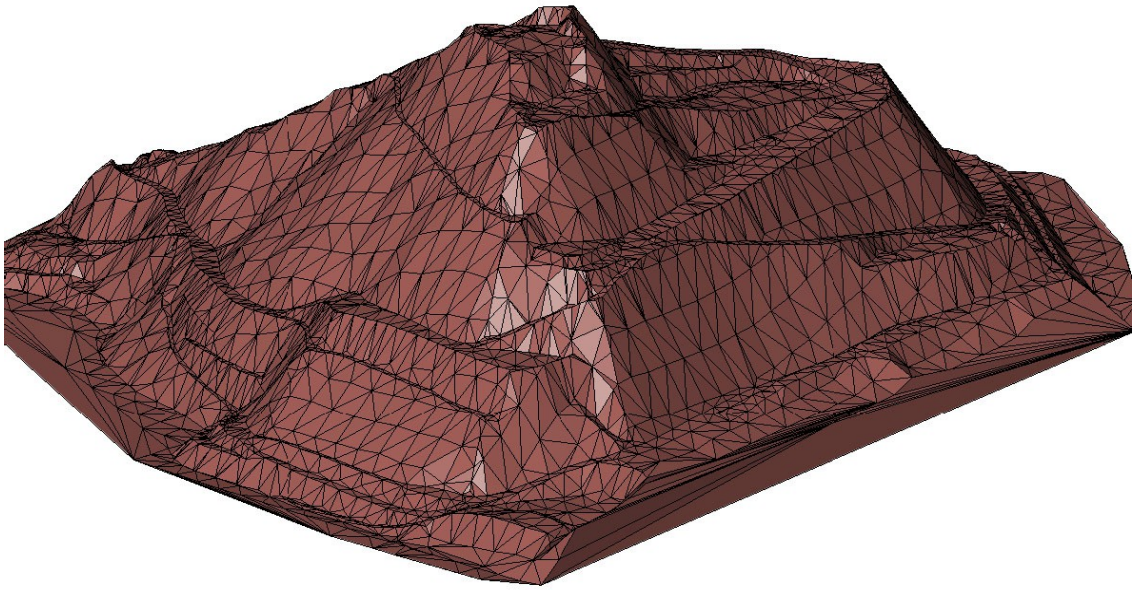
```
template < class Geometry >
class Delaunay_triangulation_2 {
    Geometry::Orientation orientation;
    Geometry::In_circle in_circle;

    void insert(Geometry::Point t) {
        ...
        if(in_circle(p,q,r,t)) {...}
        ...
        if(orientation(p,q,r){...}
    }
};
```

Généricité dans CGAL

Sans conversion explicite de points dans le plan

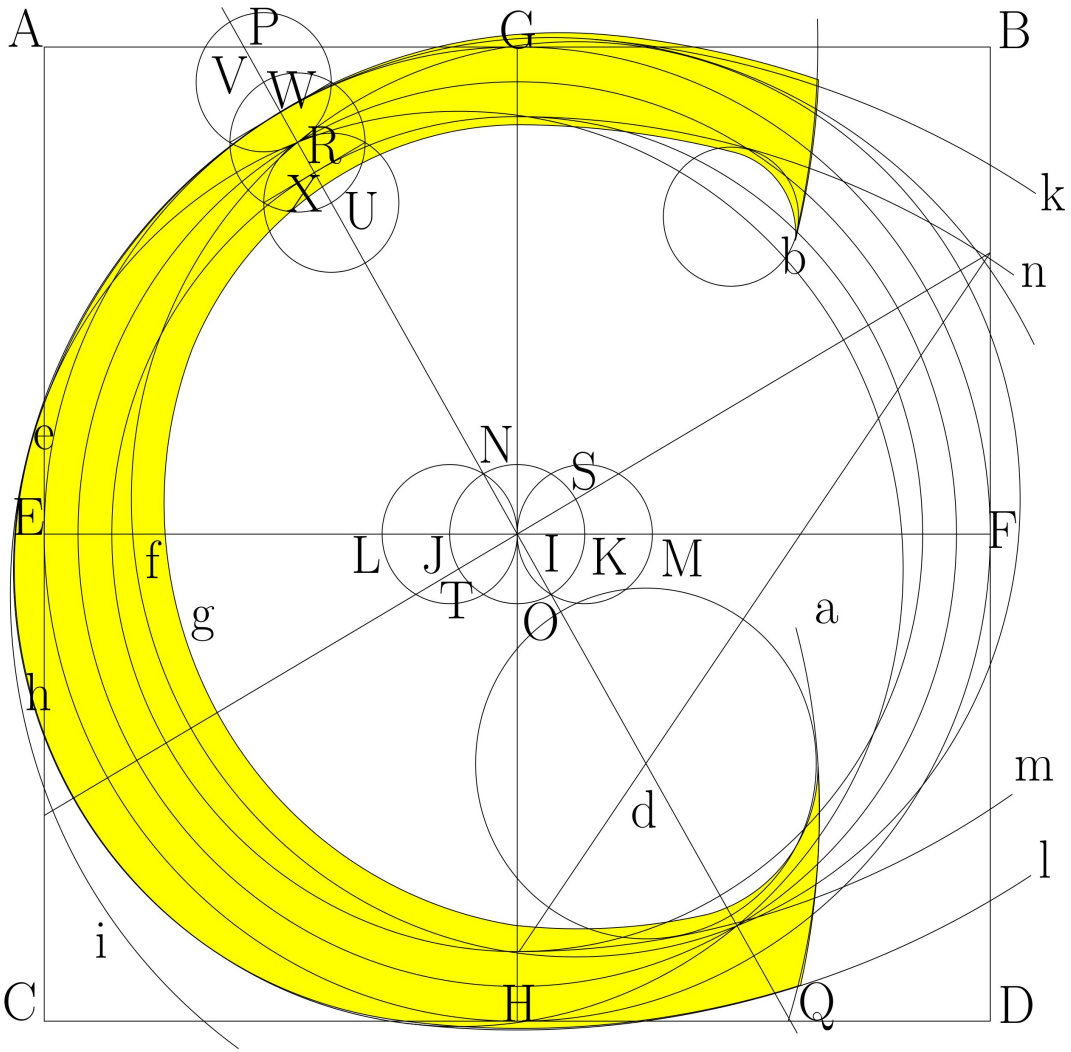
- Trianguler le terrain dans un plan xy
- Trianguler les faces d'un Polyhèdre



Courtesy: IPF,Vienna University of Technology & Inpho GmbH

Boost Graph Library (BGL)

- Collection d'algorithmes de graphes:
  plus courts chemins, arbres couvrants...
- Architecture
  - algorithmes indépendants des structures de données
  - interface fine (graph_traits et fonctions libres)
- BGL et CGAL
  - Interface pour triangulations, arrangements et HDS
  - Extension aux graphes induits par la notion de faces (algorithmes génériques sur les surfaces polyédriques)



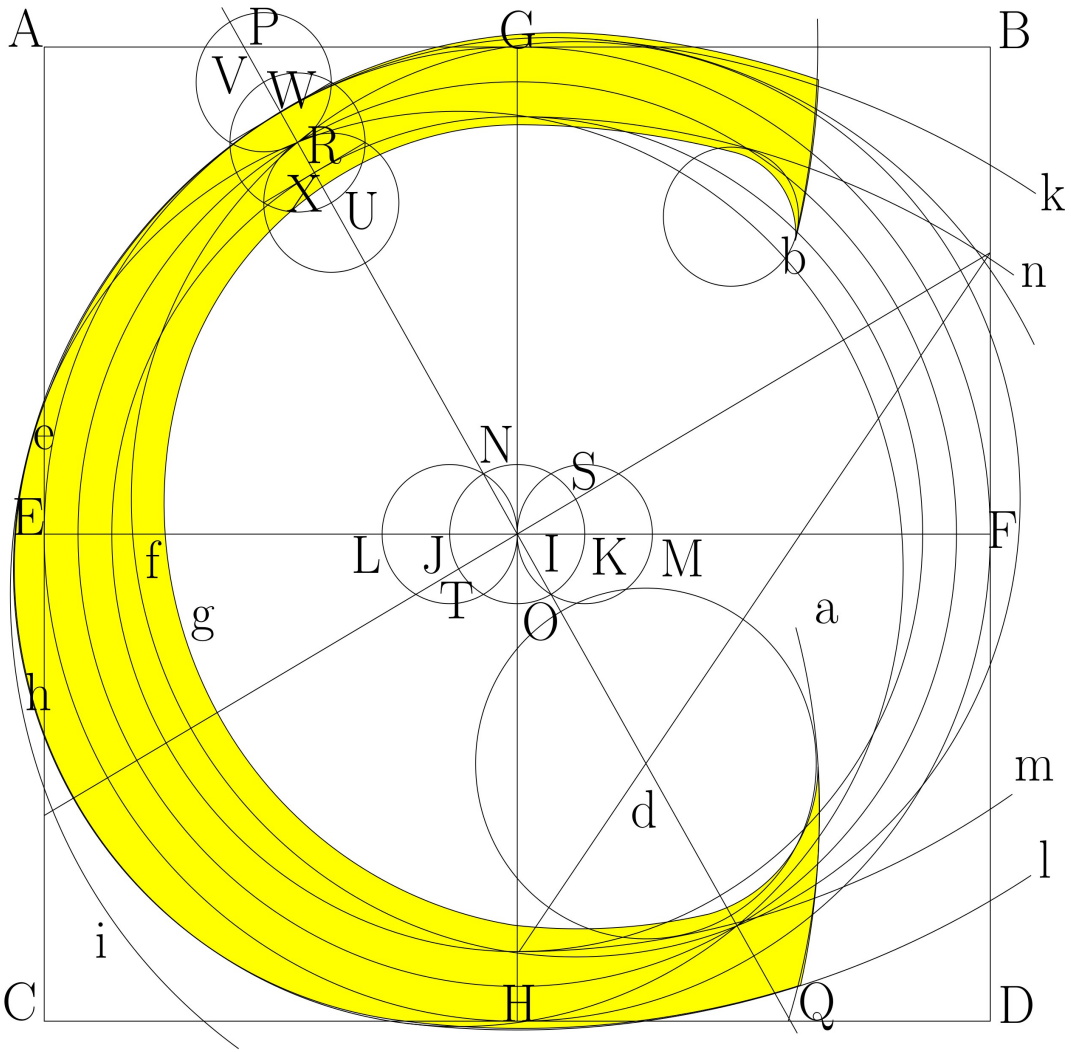
Qualité et Robustesse

Ingénierie classique

- Documentation
- Suite de programmes de tests (unitaires, régression)
- Vérification fréquente des invariants (assertions, programmation défensive)

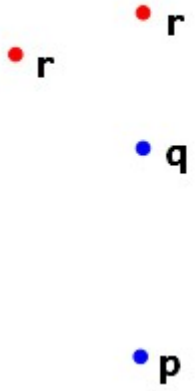
Vérification formelle automatique

- Quelques algorithmes : systèmes de preuves formelles comme COQ
- Prédicats filtrés : propagation d'erreurs d'arrondis numériques, via COQ+Gappa
- Très difficile en général

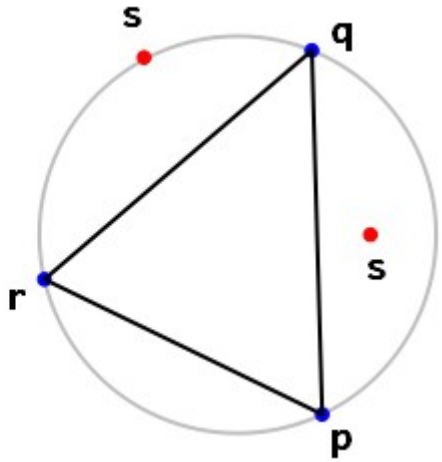


Calcul Géométrique Exact

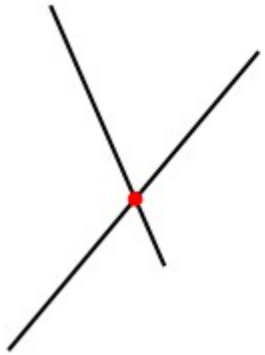
Prédicats et Constructions



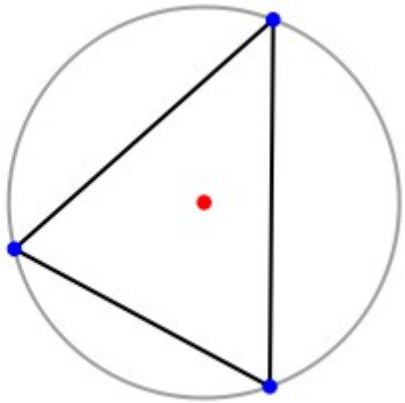
orientation



in_circle



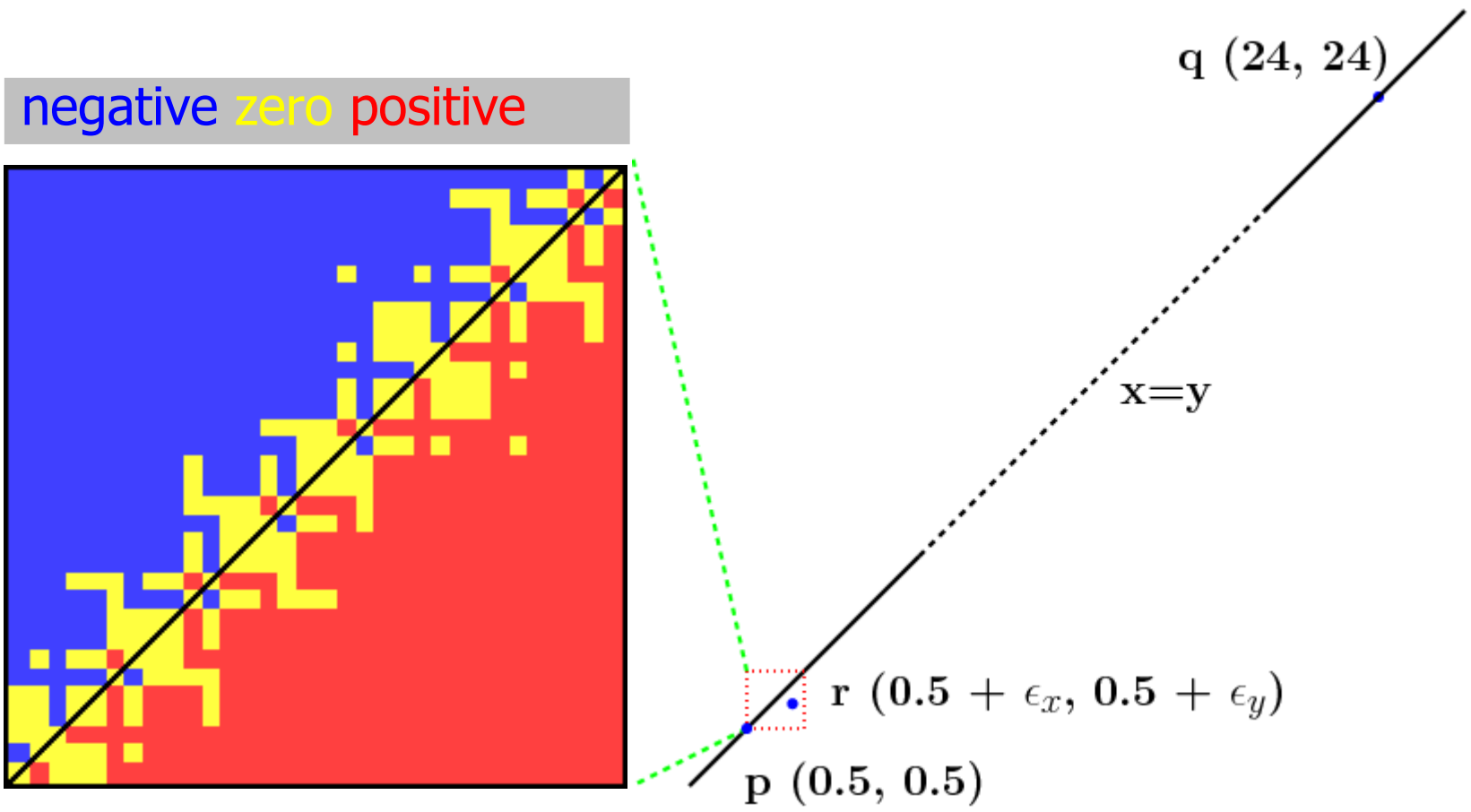
intersection



circumcenter

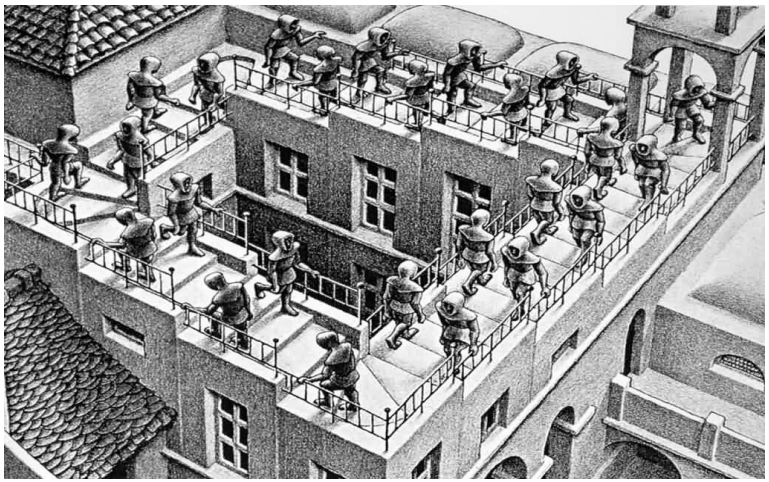
Le problème avec les flottants

orientation(p,q,r) = sign($(p_x-r_x)(q_y-r_y)-(p_y-r_y)(q_x-r_x)$)



Problème de non-robustesse

- Utilisation naïve de l'arithmétique flottante fait que les algorithmes géométriques:
  - Produit une sortie [légèrement] fausse
  - Bloque sur un invariant non-vérifié
  - Boucle infiniment



- Il y a une différence :
  - Géométrie en théorie
  - Géométrie avec arithmétique flottante

Arithmétique d'intervalles

- Encadrement garanti d'une valeur exacte dans un intervalle $[\underline{x} ; \overline{x}]$
- Toutes opérations + - * /, exemple :
  $a+b : [\underline{a} \pm \underline{b} ; \overline{a} + \overline{b}]$

- 2-10 fois plus lent que l'arith. flottante
- Les intervalles peuvent grossir beaucoup
- Nouveau standard IEEE-1788

Analyse statique d'erreurs d'arrondis

- Pour les prédicats simples comme orientation() qui sont des signes de polynomes
- À partir d'une borne sur les entrées, on peut déduire l'erreur maximale commise sur la valeur finale.

- <2 fois plus lent que l'arith. flottante

Arithmétique Multi-précision

- Modèle Real-RAM en pratique ?
- Nombres entiers exacts
  12345678901234567890123456789O

- Rationnels : p/q
- Flottants multiprécision : m.2^e

- Complexité minimum O(log n)
- Bibliothèques existantes : GMP, LEDA...

Nombres algébriques

- Exemple : comparaison d'abscisses d'intersections de cercles
- sign(sqrt(2)^2-2) ?

- Approximation numérique avec borne de séparation [CORE, LEDA].

- Méthodes algébriques de manipulation de [systèmes de] polynomes [RS]

Prédicats filtrés

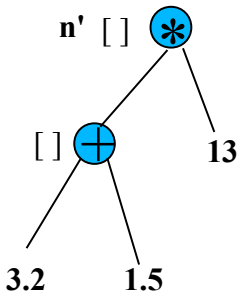
Generic functor adaptor Filtered_predicate<P>

```
try {
    évalue le prédicat P
    avec arithmétique d'intervalles;
} catch(UncertaintyException e) {
    évalue le prédicat P
    avec arithmétique exacte multi-précision;
}
```

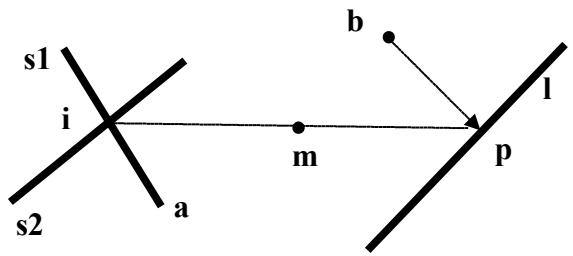
Constructions filtrées

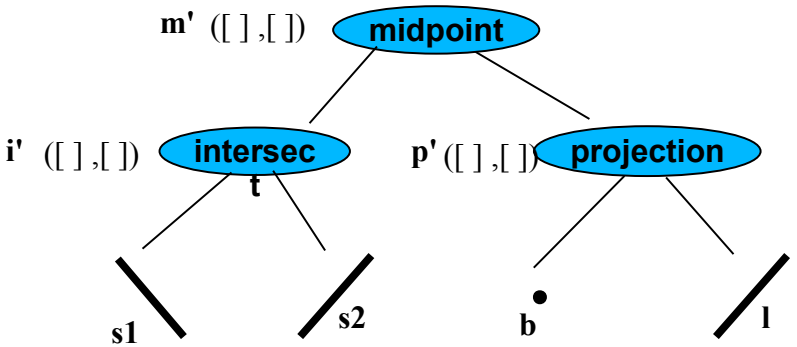
Nombre paresseux = intervalle +
arbre d'expression

$(3.2 + 1.5) * 13$



Objet paresseux = objet approché +
arbre d'opérations géométrique





Test pouvant déclencher une ré-évaluation exacte :

if (n' < m')

if (collinear(a',m',b'))

Succès et Limitations

- Robustesse numérique grâce à une boîte noire
- Surcoût en temps de calcul des prédicats exacts est raisonnable, par exemple, 10% pour triangulation Delaunay 3D
- Limitations du Calcul Géométrique Exact
  - Arrondi flottant préservant la topologie non-trivial
  - Profondeur des constructions doit être raisonnable
  - Ne peut pas gérer les fonctions trigonométriques

Conclusion

- CGAL, une grande collection d'algorithmes et structures de données géométriques
- Emphase sur la qualité : robustesse, efficacité, généricité (adaptable et extensible), documentation
- CGAL, un projet à succès, issu de la recherche académique Européenne
- http://www.cgal.org/