
PostGIS – Nouvelles fonctionnalités

▪ Licence GNU FDL - Version 1.2

- ❖ ST_SimplifyPreserveTopology
- ❖ AsGeoJson
- ❖ Prepared Topology Query

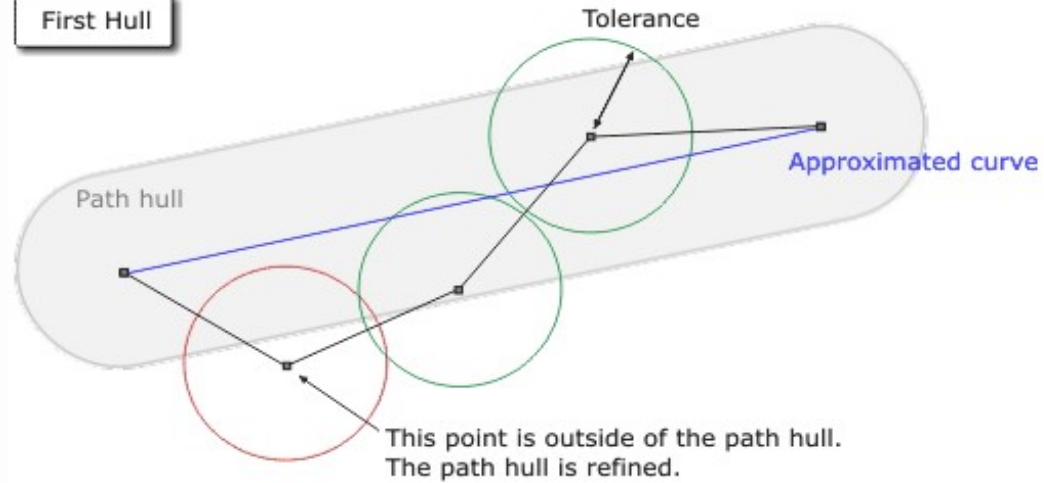


Algo Douglas Peuker

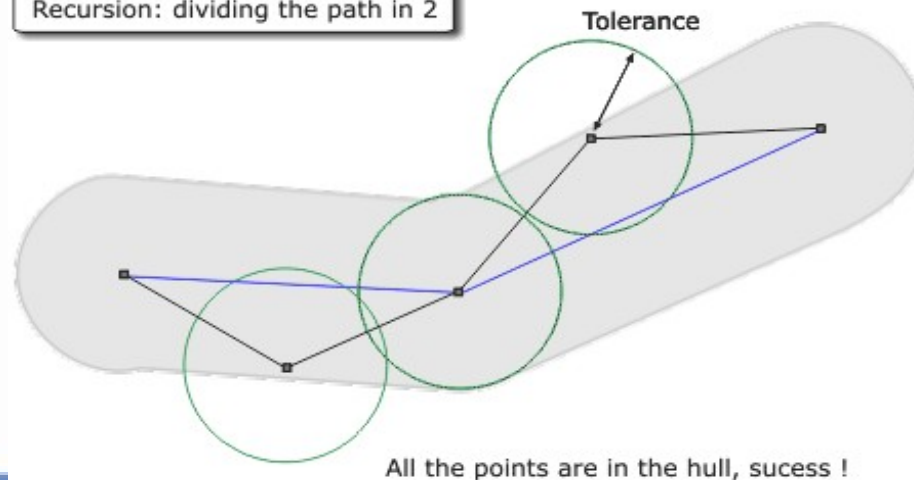
Original curve



First Hull



Recursion: dividing the path in 2



```
SELECT summary(the_geom) FROM geofla WHERE  
nom_dept='FINISTERE';
```

```
SELECT summary(simplify(the_geom, 50)) FROM  
geofla WHERE nom_dept='FINISTERE';
```

```
SELECT isvalid(simplify(the_geom, 50)) FROM  
geofla WHERE nom_dept='FINISTERE';
```

Simplify 'classique' avec forte généralisation

```
SELECT summary(the_geom) FROM geofla WHERE  
nom_dept='FINISTERE';
```

```
SELECT summary(simplify(the_geom, 200)) FROM  
geofla WHERE nom_dept='FINISTERE';
```

```
SELECT isvalid(simplify(the_geom, 200)) FROM  
geofla WHERE nom_dept='FINISTERE';
```



Erreurs présentes lorsque

- ❖ Polygones multiples (et/ou avec plusieurs ring)
- ❖ Généralisation suffisamment forte
- ❖ => Génère des polygones qui s'intersectent mutuellement !





ST_SimplifyPreserveTopology

```
SELECT
    summary(ST_SimplifyPreserveTopology(the_geom, 200))
FROM
    geofla WHERE nom_dept='FINISTERE';
```

```
SELECT
    isvalid(ST_SimplifyPreserveTopology(the_geom, 200))
FROM
    geofla WHERE nom_dept='FINISTERE';
```



ST_SimplifyPreserveTopology: perspectives

- ❖ Dispo dans version 1.3.3 PostGIS
- ❖ A utiliser de manière préférentielle à `simplify()` pour des (multi)polygones
- ❖ Si besoin de préserver la topologie entre features, pas encore possibilité de le gérer à la volée (modèle spaghetti)
 - ❖ Moulinette de Laurent Pierre disponible ([site cartoweb.org addons](http://site.cartoweb.org/addons))



- ❖ Disponible dans le SVN (prochaine version 1.4.x)
- ❖ Implémentation GeoJson 1.0 rc6
 - ❖ Cf <http://wiki.geojson.org/GeoJSON_draft_version_6>
- ❖ Géométries supportées
 - ❖ (MULTI)POINT
 - ❖ (MULTI)LINE
 - ❖ (MULTI)POLYGON
 - ❖ GEOMETRYCOLLECTION

- ❖ ST_AsGeoJson([version], geometry, [precision], [options])
 - ❖ version: toujours 1
 - ❖ Geometry: la geom (cf types supportés)
 - ❖ precision: precision (nombre de digits désirés, default 15)
 - ❖ options:
 - ❖ 0 pas d'option (défaut)
 - ❖ 1 GeoJson CRS (ne le place pas si valeur inconnue)
 - ❖ 2 GeoJson BBOX (le recalcule si pas contenu dans la geom)
 - ❖ 3 GeoJson CRS and BBOX



ST_AsGeoJson, exemple

```
bar=# SELECT ST_AsGeoJson(GeomFromEWKT('SRID=4326;POINT(1 1)'), 0);
```

```
st_asgeojson
```

```
-----
```

```
{"type": "Point", "coordinates": [1, 1]}
```

```
SELECT ST_AsGeoJson(GeomFromEWKT('LINESTRING(1 1, 2 2, 3 3, 4 4)'),  
0, 2);
```

```
st_asgeojson
```

```
-----
```

```
-----  
{"type": "LineString", "bbox": [1, 1, 4, 4], "coordinates": [[1, 1], [2, 2],  
[3, 3], [4, 4]]}
```



-
- ❖ Utilisation de GeoJson directement dans OL:
 - ❖ Cf <<http://openlayers.org/dev/examples/vector-formats.html>>
- ❖ Utilisation de WS light couplé à PostGIS
- ❖ Appel direct de MapFish Server
 - ❖ Quid d'un bench sur perf ?
- ❖ ...



- ❖ GEOS gère opérations topologiques (intersects, contains, ...) et le calcul (intersection, buffer, simplify, ...)
- ❖ Comparaisons « 1 à 1 » ou « 1 à n » parmi les plus fréquentes
- ❖ Pour optimiser les comparaisons « 1 à n »: système de mise en cache (prepared) à partir de GEOS 3.1 et PostGIS 1.3.3





ST_intersects avec et sans prepared cache

```
SELECT nom_dept FROM geofla
```

```
WHERE
```

```
  ST_intersects(
```

```
    (SELECT the_geom FROM geofla WHERE nom_dept='SAVOIE'),  
    the_geom  
  );
```

```
SELECT nom_dept FROM geofla
```

```
WHERE
```

```
  ST_intersects(
```

```
    (SELECT the_geom FROM geofla WHERE nom_dept='SAVOIE'),  
    the_geom,  
    1  
  );
```

