



PostGIS – Nouvelles fonctionnalités

Licence GNU FDL - Version 1.4



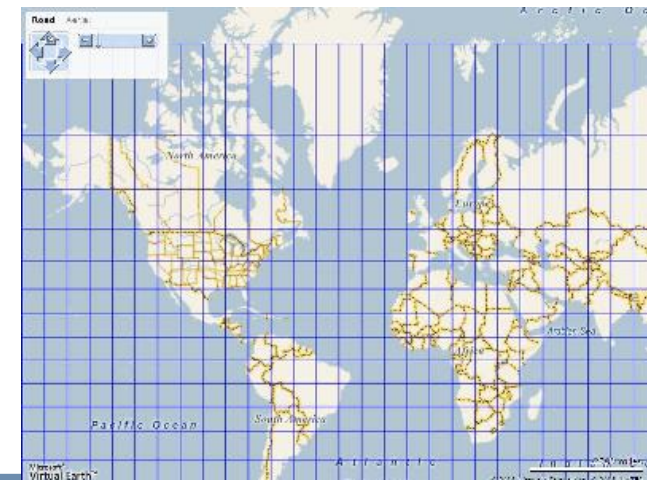
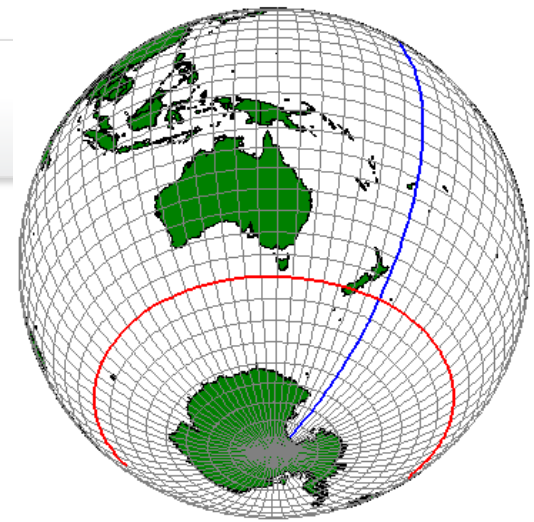
Plan de la présentation

- Nouveau type GEOGRAPHY
- Nouvelles fonctions
- ST_SimplifyPreserveTopology
- AsGeoJson
- Prepared Topology Query



Le type GEOGRAPHY

- Nouveauté depuis PostGIS 1.5
- Gestion des coordonnées latitude, longitude
- Intègre un modèle sphérique de la terre
- Permet de gérer des données globales
- Enregistrement automatique des métadonnées de la table
- Types supportés:
 - POINT, LINESTRING, POLYGON,
 - MULTIPOINT, MULTILINESTRING, MULTIPOLYGON, GEOMETRYCOLLECTION
- GEOS non disponible
- Support partiel: nouvelles fonctions à venir





Le type GEOGRAPHY

- ST_Area
- ST_AsBinary
- ST_AsGML
- ST_AsGeoJSON
- ST_AsKML
- ST_AsSVG
- ST_AsText
- ST_Buffer
- ST_CoveredBy
- ST_Covers
- ST_Dwithin
- ST_Distance
- ST_GeogFromText
- ST_GeogFromWKB
- ST_GeographyFrom
- ST_Intersection
- ST_Intersects
- ST_Length



Le type GEOGRAPHY

- Création d'une table:

```
CREATE TABLE global_points (  
  id SERIAL PRIMARY KEY,  
  name VARCHAR(64),  
  location GEOGRAPHY(POINT,4326));
```

- Seul SRID supporté: EPSG:4326
- Insertion:

```
INSERT INTO global_points (name, location)  
VALUES ( 'Town', ST_GeographyFromText( 'SRID=4326;POINT(-110 30)' ));
```



Le type GEOGRAPHY

- Création d'un index spatial:

```
CREATE INDEX global_points_gix  
ON global_points USING GIST ( location );
```

- Utilisation d'unités métriques (distances, aires):

```
SELECT name FROM global_points  
WHERE ST_DWithin(location,  
ST_GeographyFromText('SRID=4326;POINT(-110 29)'), 1000000);
```

```
SELECT ST_Distance(  
    'LINESTRING(-122.33 47.606, 0.0 51.5)::geography',  
    'POINT(-21.96 64.15):: geography'  
);
```



Nouvelles fonctions

&&&	Loader_Generate_Census_Script	ST_AsTIFF
<#>		ST_AsX3D
<	Loader_Generate_Script	ST_Aspect
AddEdge	Missing_Indexes_Generate_Script	ST_Band
AddFace		ST_BandsNoData
AddNode	Polygonize	ST_Clip
AddRasterConstraints	Reverse_Geocode	ST_CollectionHomogenize
AsGML	ST_3DClosestPoint	ST_ConcaveHull
CopyTopology	ST_3DDFullyWithin	ST_Count
DropRasterConstraints	ST_3DDWithin	ST_CreateTopoGeo
Drop_Indexes_Generate_Script	ST_3DDistance	ST_Distinct4ma
Drop_State_Tables_Generate_Script	ST_3DIntersects	ST_FlipCoordinates
Geocode_Intersection	ST_3DLongestLine	ST_GDALDrivers
GetEdgeByPoint	ST_3DMaxDistance	
GetFaceByPoint	ST_3DShortestLine	
GetNodeByPoint	ST_AddEdgeModFace	
GetNodeEdges	ST_AddEdgeNewFaces	
GetRingEdges	ST_AsGDALRaster	
GetTopologySRID	ST_AsJPEG	
Get_Tract	ST_AsLatLonText	
Install_Missing_Indexes	ST_AsPNG	
	ST_AsRaster	



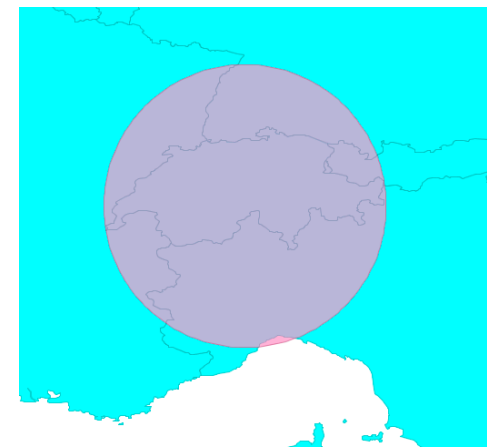
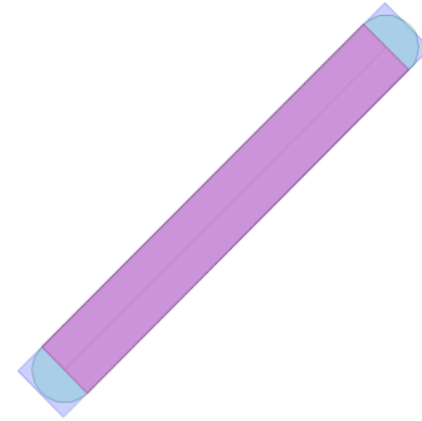
Nouvelles fonctions (suite)

ST_GeomFromGeoJSON	ST_NewEdgeHeal	ST_Snap
ST_GetFaceEdges	ST_Node	ST_SnapToGrid
ST_HasNoBand	ST_NumPatches	ST_Split
ST_HillShade	ST_OffsetCurve	ST_StdDev4ma
ST_Histogram	ST_PatchN	ST_Sum4ma
ST_InterpolatePoint	ST_Quantile	ST_SummaryStats
ST_IsValidDetail	ST_Range4ma	ST_Transform
ST_IsValidReason	ST_Reclass	ST_UnaryUnion
ST_MakeLine	ST_RelateMatch	ST_Union
ST_MakeValid	ST_RemEdgeModFace	ST_ValueCount
ST_MapAlgebraExpr	ST_RemEdgeNewFace	TopoElementArray_Agg
ST_MapAlgebraExpr	ST_RemoveRepeatedPoints	TopoGeo_AddLineString
ST_MapAlgebraFct	ST_Resample	TopoGeo_AddPoint
ST_MapAlgebraFct	ST_Rescale	TopoGeo_AddPolygon
ST_MapAlgebraFctNgb	ST_Reskew	TopologySummary
ST_Max4ma	ST_SameAlignment	Topology_Load_Tiger
ST_Mean4ma	ST_SetBandIsNoData	toTopoGeom
ST_Min4ma	ST_SharedPaths	
ST_ModEdgeHeal	ST_Slope	



Nouvelles fonctions

- **ST_Buffer**: nouveaux types de connection, de fin
- **ST_ClosestPoint**: retourne le point d'une geometry le plus proche d'une autre
- **ST_DFullyWithin**: True si une geometry est à une distance minimale d'une autre
- **ST_dumpPoints**: Extrait tous les points d'une geometry sous forme d'un set
- **ST_geomFromGML**, **ST_geomFromKML**: conversion depuis le format GML, KML
- **ST_HausdorffDistance**: Similitude de deux geometry
- **ST_minimumBoundingCircle**: retourne le plus petit cercle qui contient la geometry





Nouvelles fonctions

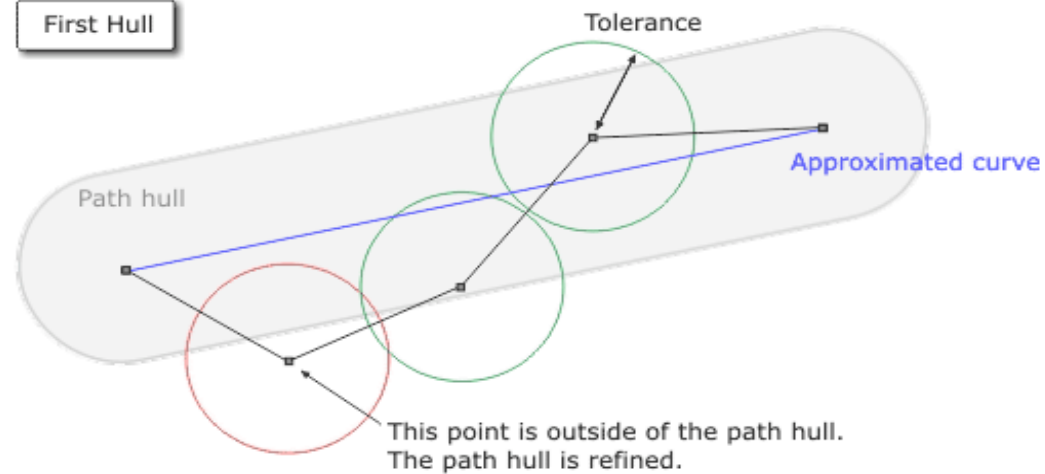
- **st_shortestLine, st_longestLine**: ligne la plus courte, la plus longue, entre deux geometry
- **st_maxDistance**: distance max entre deux geometry
- **st_isValidReason**: donne la raison de l'invalidité d'une geometry
- **st_makeReason**: corrige une géométrie invalide
- **st_addMeasure**: ajoute une dimension « Measure » interpolée à la geometry (LINESTRING ou MULTILINESTRING)
- ...

St_simplify

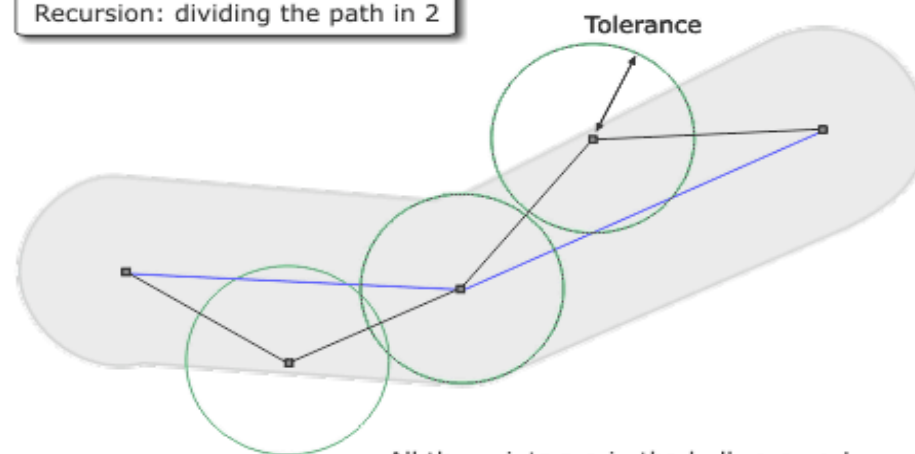
Original curve



First Hull



Recursion: dividing the path in 2

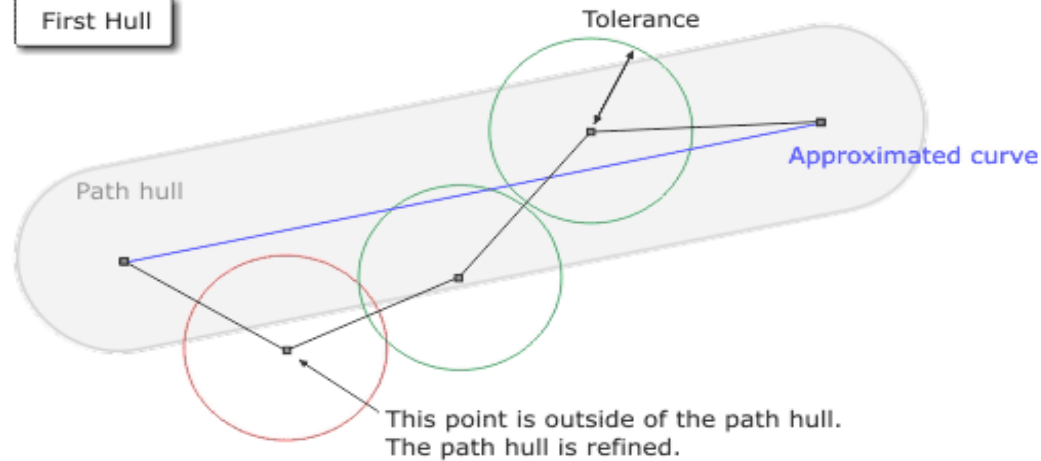


St_simplify: Douglas Peucker

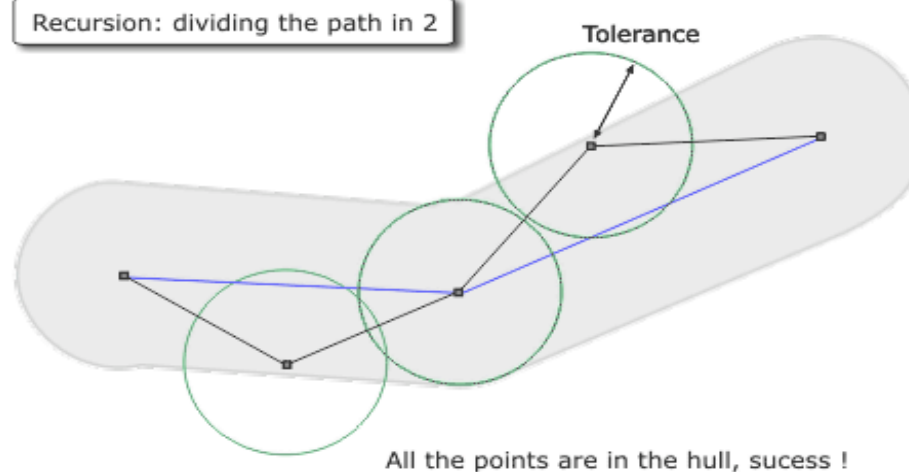
Original curve



First Hull



Recursion: dividing the path in 2





Simplify 'classique'

```
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';
```

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, utiliser PostGIS Topology



ST_AsGeoJson

- Géométries supportées
 - (MULTI)POINT
 - (MULTI)LINE
 - (MULTI)POLYGON
 - GEOMETRYCOLLECTION



ST_AsGeoJson, prototype

- 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, example

```
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]] }



ST_AsGeoJson, perspectives

- Utilisation de GeoJson directement dans OpenLayers:
 - Cf <http://openlayers.org/dev/examples/vector-formats.html>
- Utilisation de WS light couplé à PostGIS
- Appel direct de MapFish Server
- ...