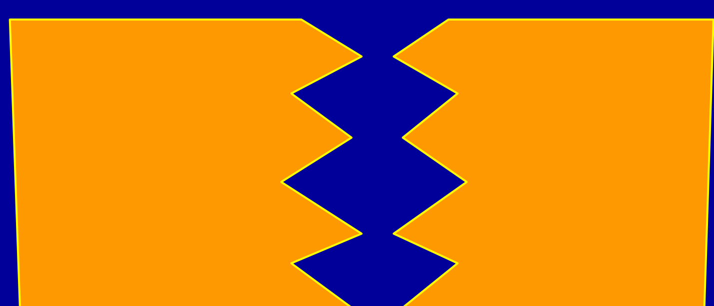
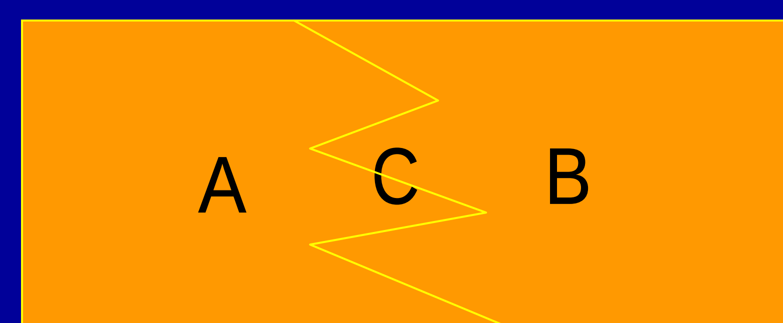
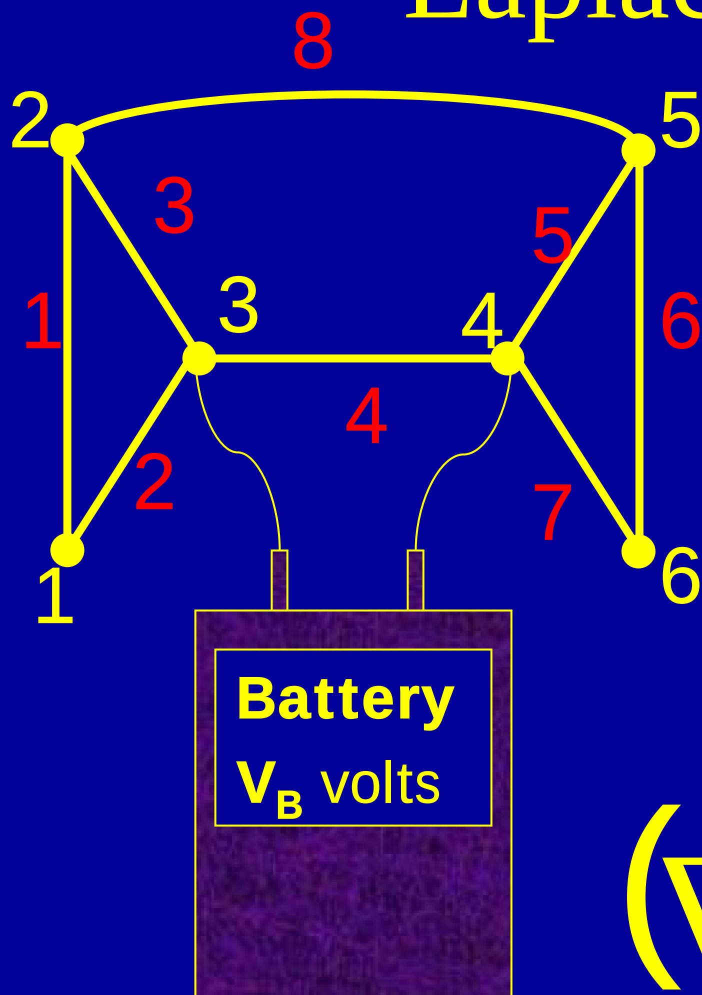
Spectral Partitioning: One way to slice a problem in half





Laplacian of a Graph ∇_G^2



$$\begin{bmatrix} 2 & -1 & -1 & & & \\ -1 & 3 & -1 & & -1 & \\ -1 & -1 & 3 & -1 & & \\ & & -1 & 3 & -1 & -1 \\ & -1 & & -1 & 3 & -1 \\ & & & -1 & -1 & 2 \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \\ v_3 \\ v_4 \\ v_5 \\ v_6 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 1 \\ -1 \\ 0 \\ 0 \end{bmatrix}$$

$(\nabla_G^2)_{ii} =$ degree of node i

$(\nabla_G^2)_{ij} =$ -1 for edges i, j

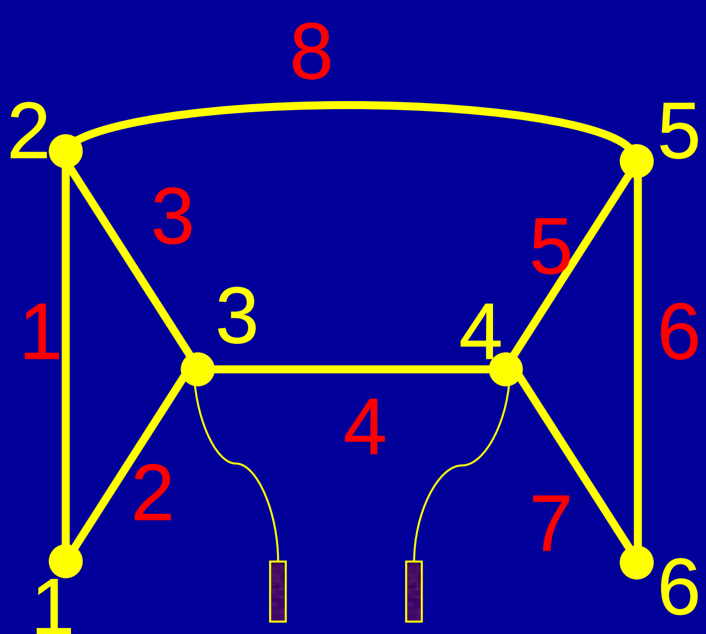
Edge-Node Incidence Matrix

$$\nabla_G^2 =$$

2	-1	-1			
-1	3	-1		-1	
-1	-1	3	-1		
		-1	3	-1	-1
	-1		-1	3	-1
			-1	-1	2

$$M_G =$$

	1	2	3	4	5	6
1	1	-1				
2	1		-1			
3		1	-1			
4			1	-1		
5				1	-1	
6					1	-1
7				1		-1
8		1			-1	



$$\nabla_G^2 = M_G^T M_G$$

Spectral Partitioning

Express partition problem with linear algebra!

Partition vector: $x_i = \pm 1$ denotes i's partition.

$$\sum_{i=1}^{n} (M_G x)_i^2 = (M_G x)^T (M_G x) = x^T M_G^T M_G x = x^T \nabla_G^2 x$$

$$x^T \nabla_G^2 x = 4(\#\text{ cross partition edges})$$

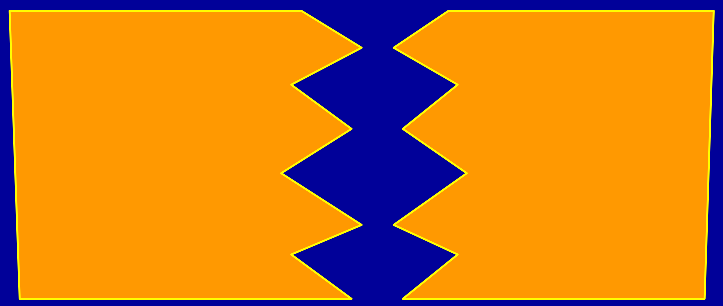
Goal is to minimize this for $x_i = \pm 1$ and $\Sigma x_i = 0$.

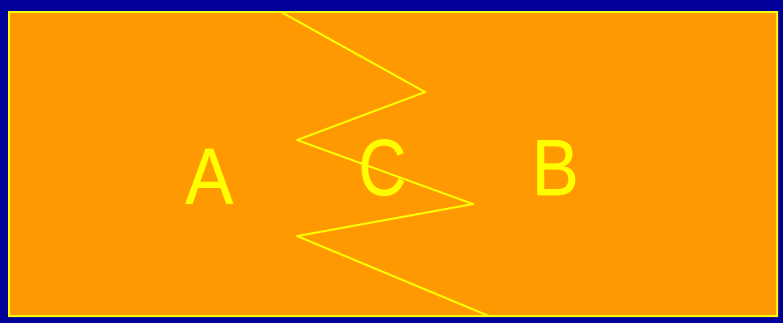
Bounded above by minimizing over ball $\Sigma x_i^2 = n$.

Solve as an eigenvalue problem!

Geometric Mesh Partitioning:

The Miller, Teng, Thurston, Vavasis Algorithm





Graph Partitioning

- Division of graph into subgraphs with goal of minimizing communication and maximizing load balance.

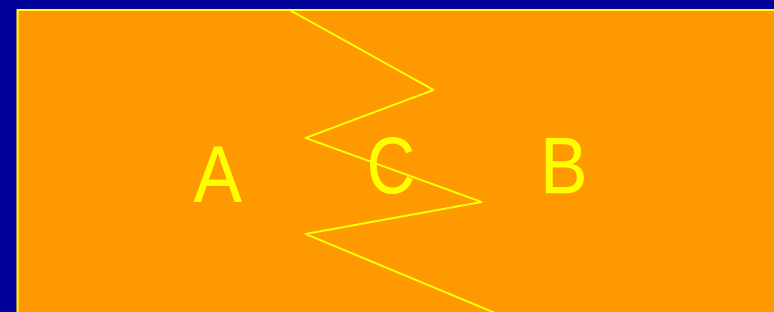
Geometric Methods

- Uses not only the graph (= combinatorial information) but also geometrical coordinates (x,y) or (x,y,z) for the nodes.

Edge Separator

Remove Separator Edges

Vertex Separator

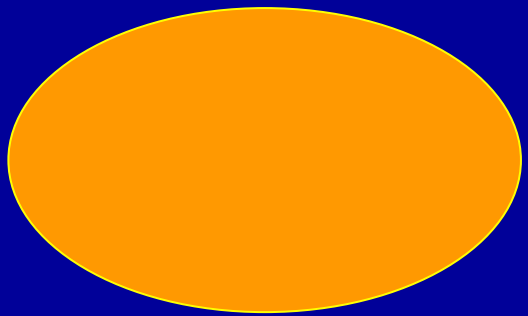


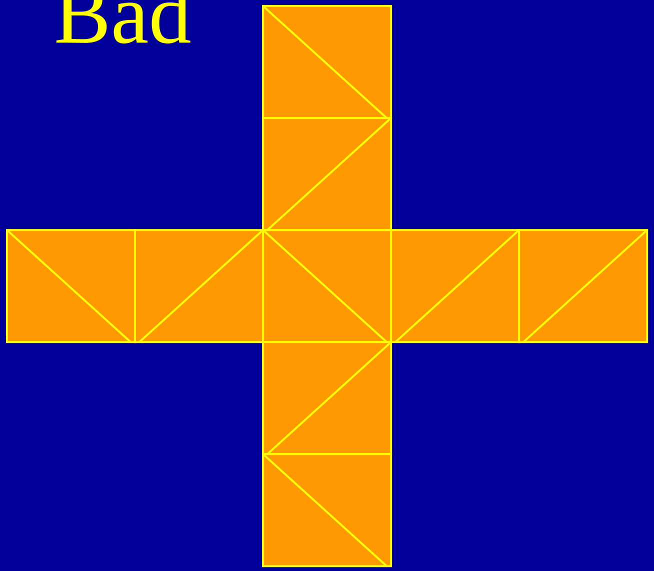
Nodes = $A \cup B \cup C$

No edges between A and B

Simple Geometric Partitioner: Coordinate Bisection

- 1. Compute median of x and y coordinates
- 2. Count edges along x=mean(x) and y=mean(y)
- 3. Use cut which minimizes the two
- OK for   Bad



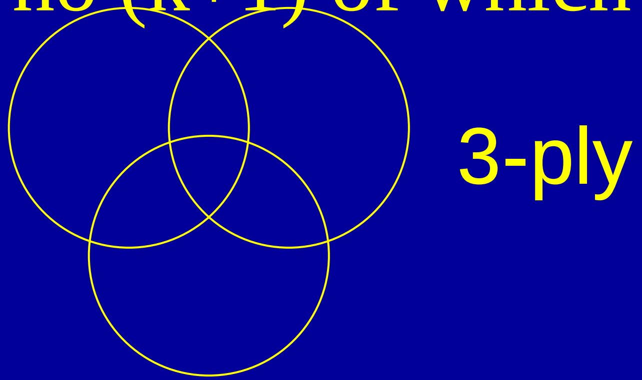


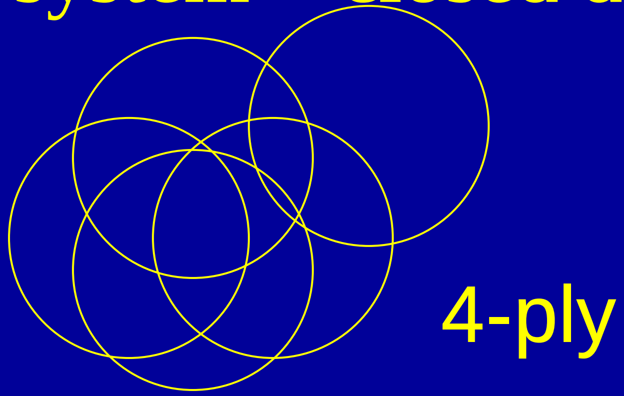
Theory vs. Practice

- Random Graphs may have lousy separators
- Practical Numerical Meshes often have good separators
- Fact: For numerical reasons, meshes have good aspect ratios
- In 2-d:
- A_1=longest edge/shortest edge
- A_2=circumsphere/inscribed sphere
- A_3=diameter / volume^(1/d)

Need Theoretical Class of Good Graphs

- Defn: k-ply neighborhood system = closed disks no (k+1) of which overlap





- A (1,k) overlap graph for a k-ply system =graph obtained by connecting centers of intersecting circles
- (α,k) overlap graph = connect centers if $\alpha D_i \cap D_j \neq \emptyset$ ($\alpha < 1$)

Why is this useful?

- Intuitively: Good overlap graph = bounded aspect ratio
- Theoretical Theorems best proved about overlap graphs. Example:
- Geometric Separator Theorem:

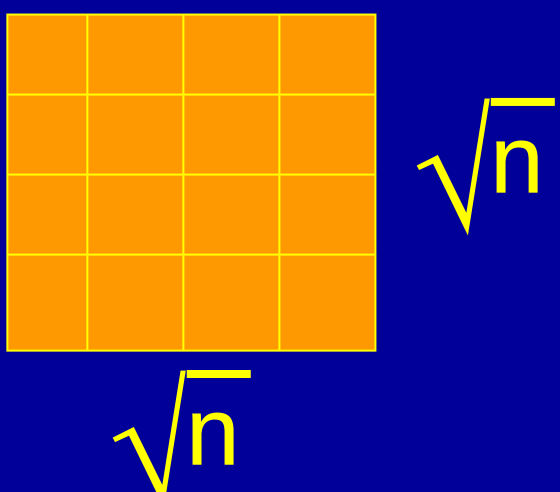
Geometric Separator Theorem

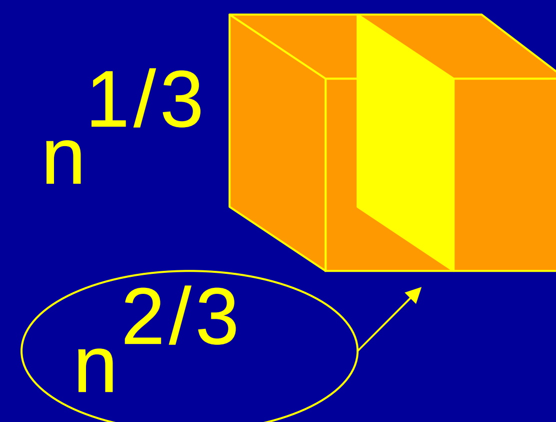
If G=(α,k) overlap graph in d dimensions,

there exists a vertex separator with

O(α k^(1/d) n^((d-1)/d)) vertices.

In English: All good overlap graphs behave as if they are cubes. A d-cube with n vertices has a separator of size n^((d-1)/d).



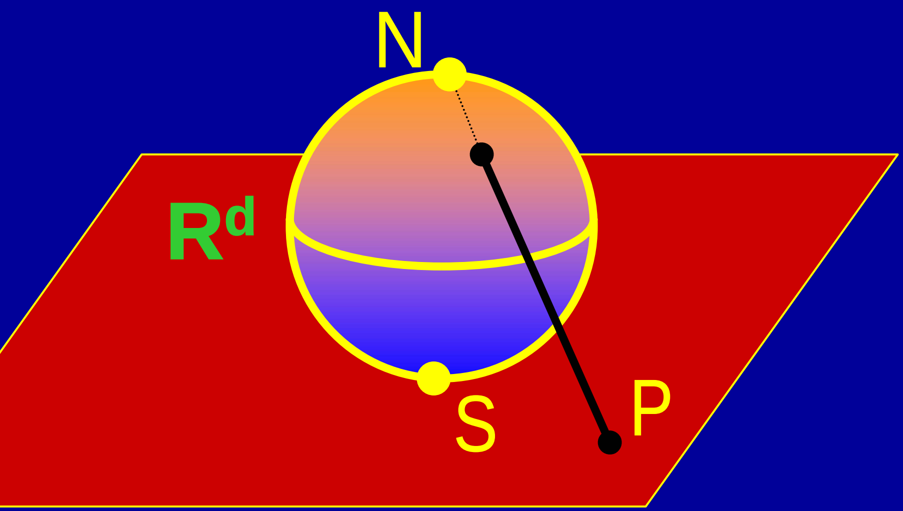


Miller, Teng, Thurston, Vavasis algorithm for finding separator

1) Stereographic Projection
2) Find centerpoint
3) Conformal Map
4) Find Great Circle
5) Project Back
6) Create Separator

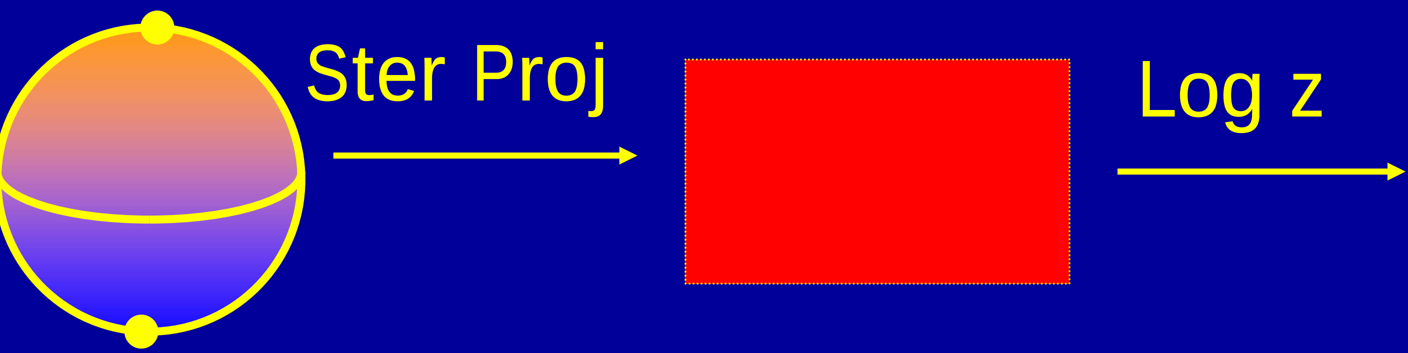
Details ...

Step 1:Stereographic Projection



Project P to sphere along the line to the north pole.

Mercator Map



Not cylindicrical projection!

Step 2: Find Centerpoint

Definition: A Centerpoint is a point C such that every hyperplane through C roughly divides the points evenly.

Theorem: Every finite set has a centerpoint -- may be found my linear programming (not practical). Later: A practical heuristic.

Step 3: Conformal Map to move centerpoint to center of sphere

Why? Increases chances of a good cut.

Rotate and Dilate:

Rotate: centerpoint to $(0,0,r)$

Dilate: centerpoint to $(0,0,0)$

Steps 4 through 6

4) Cut with a random great circle.

5) Stereographic projection back to the plane from the sphere.

6) Convert the circle in the plane to a separator.

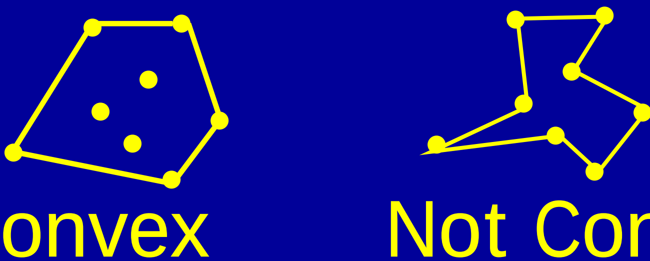
DEMO!

Centerpoint Computation

Heuristic runs in linear time by computing Radon points.

Definition: q is a Radon point of a set of points **P** if $\mathbf{P}=\mathbf{P}_1 \cup \mathbf{P}_2$ (disjoint union) and q is in both the convex hull of $\mathbf{P}_1$ and $\mathbf{P}_2$.

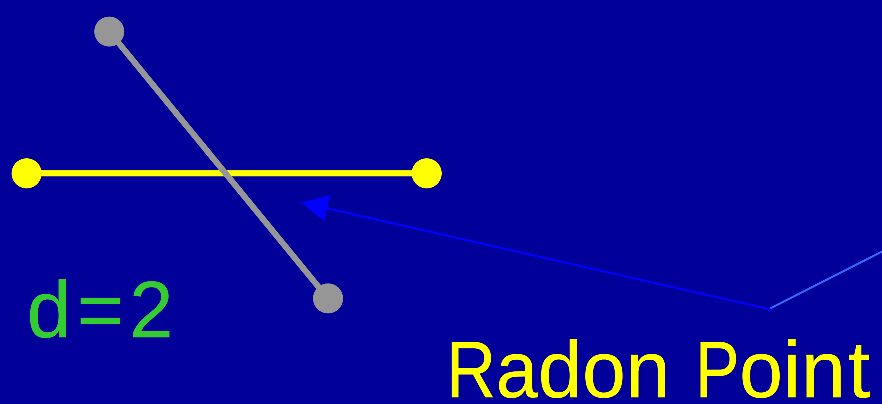
Convex hull -- smallest convex polygon containing the set.

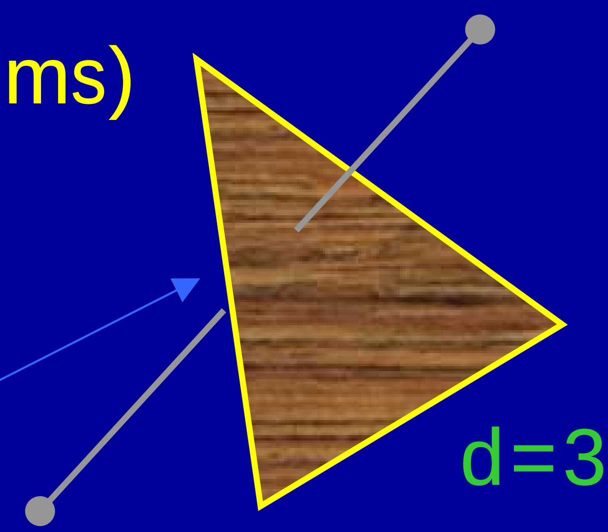


Radon Points

Definition: q is a Radon point of a set of points **P** if **P**=$\mathbf{P_1} \cup \mathbf{P_2}$ (disjoint union) and q is in both the convex hull of $\mathbf{P_1}$ and $\mathbf{P_2}$.

Examples: (d+2 points in d dims)





Computing Radon Pnts = Linear Algebra

A point P is in the convex hull of a set of points P_i if and only if it has the form $P=\Sigma\alpha_i P_i : \Sigma\alpha_i=1$, $\alpha_i \geq 0$.

Solve
$$\begin{pmatrix} P_1 & P_2 & \cdots & P_{d+2} \\ 1 & 1 & \cdots & 1 \end{pmatrix} \begin{pmatrix} \alpha_1 \\ \vdots \\ \alpha_{d+2} \end{pmatrix} = 0$$

Let $c = \sum_{\text{Pos } \alpha_i} \alpha_i = \sum_{\text{Neg } \alpha_i} -\alpha_i$.

The Radon point is then

$$\sum_{\text{Pos } \alpha_i} \alpha_i P_i / c = \sum_{\text{Neg } \alpha_i} -\alpha_i P_i / c.$$

Geometric Sampling

Select random sample of points.

Randomly replace d+2 points with Radon pt

A few tricks

1) Try a few random great circles (using normal dist)
2) Weigh the normal vector in the moment of inertia direction

METIS and Parmetis