

Representações Celulares e Esfera Geodésica

Modelagem Geométrica — IMPA 2026 — Prof. Luiz Henrique de Figueiredo
Fevereiro de 2026

1. Representações Celulares

A primeira tarefa consiste em calcular representações celulares — uniformes e adaptativas — de regiões definidas implicitamente no plano. Uma função implícita $f(x,y)$ particiona o plano em três conjuntos: **interior** ($f \leq 0$), **fronteira** ($f = 0$) e **exterior** ($f > 0$). A solução foi implementada em Python com matplotlib e numpy.

1.1 Classificação de Células via TVI

Cada célula retangular é classificada avaliando f nos seus 4 cantos. Pelo **Teorema do Valor Intermediário** (Bolzano, 1817): se f assume valores positivos e negativos nos cantos, a fronteira necessariamente cruza a célula. Se todos os valores são ≤ 0 , a célula está **dentro**; se todos > 0 , está **fora**; caso contrário, é de **fronteira**.

1.2 Enumeração Uniforme vs. Adaptativa

A **enumeração uniforme** divide o domínio numa grade regular $N \times N$ e classifica cada célula. É simples mas desperdiça resolução em regiões homogêneas. A **enumeração adaptativa** usa uma Quadtree (Finkel–Bentley, 1974): células homogêneas permanecem grandes; apenas células de fronteira são subdivididas em 4 quadrantes. Isso concentra resolução onde a geometria é interessante, seguindo o refinamento adaptativo de Suffern (1990).

1.3 Resultados — Disco

Disco com centro (0.3, 0.4) e raio 0.21 no quadrado unitário. A versão uniforme 32×32 usa 1024 células; a adaptativa, apenas 583 — 43% menos células, com resolução muito superior na fronteira.

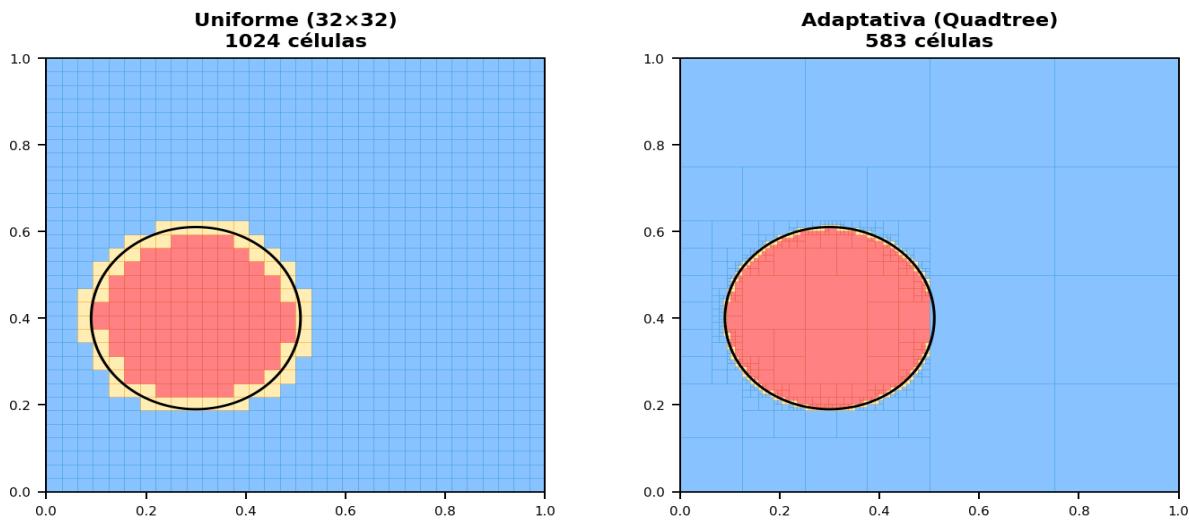


Figura 1: Disco — Uniforme (esq.) vs. Adaptativa (dir.). Vermelho = dentro, azul = fora, amarelo = fronteira.

1.4 Resultados — Parábola $y \geq x^2 + c$

Região definida por $f(x,y) = x^2 + c - y$ no domínio $[-2, 2] \times [-2, 2]$. Abaixo, $c = 0$ (enunciado) e variações.

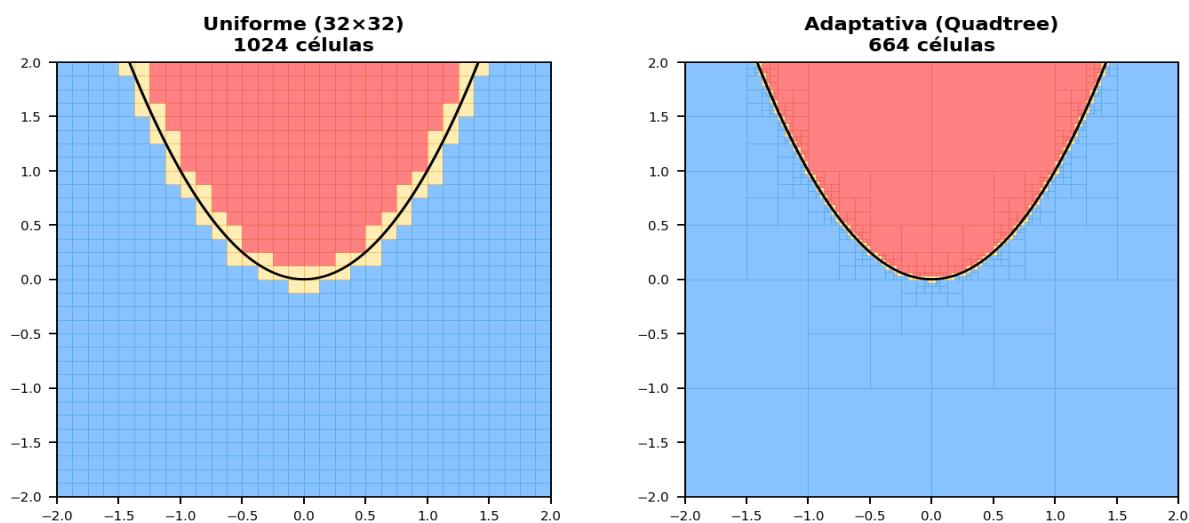


Figura 2: Parábola $c=0$ — Uniforme (esq.) vs. Adaptativa (dir.).

1.5 Evolução do Refinamento

A figura abaixo mostra como a Quadtree refina progressivamente o disco, desde profundidade 2 (16 células) até profundidade 7 (583 células). As células grandes nas regiões homogêneas demonstram a economia do método adaptativo.

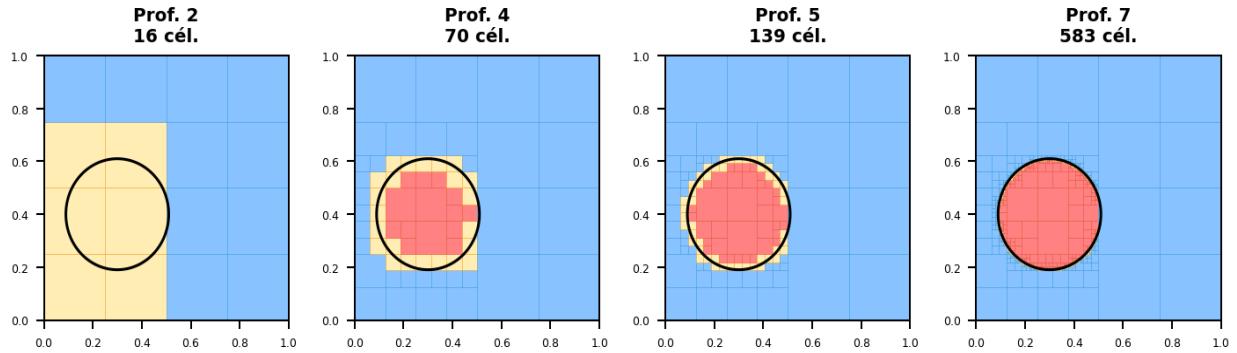


Figura 3: Evolução do refinamento adaptativo — profundidades 2, 4, 5 e 7.

1.6 Experimentação com Parâmetros

Variando o parâmetro c da parábola, a fronteira sobe ($c > 0$) ou desce ($c < 0$) no domínio, alterando a proporção entre células dentro e fora. A Quadtree se adapta automaticamente em todos os casos.

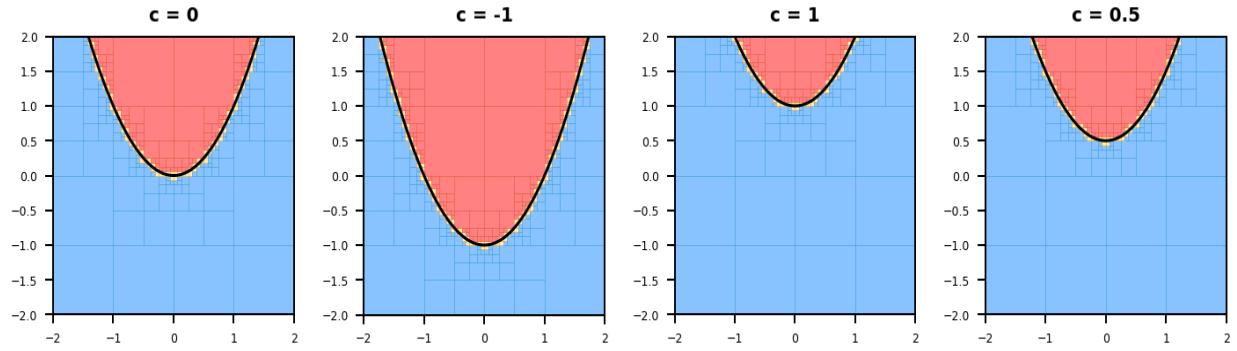


Figura 4: Representação adaptativa para $c = 0, -1, 1$ e 0.5 .

1.7 Dificuldades

A principal dificuldade inicial foi montar o ambiente em C++. Migrar para Python resolveu isso e permitiu iteração rápida com matplotlib. Conceitualmente, a classificação por cantos pode falhar quando a fronteira entra e sai da célula sem tocar nenhum canto (todos ficam com o mesmo sinal). Para as formas desta tarefa isso não ocorre, mas para formas mais complexas seria necessário usar oráculos intervalares, como discutido em aula.

2. Esfera Geodésica

A segunda tarefa consiste em gerar uma esfera geodésica no formato OBJ, partindo de um sólido platônico de faces triangulares e refinando por subdivisão com projeção na esfera unitária.

2.1 Método

Para cada triângulo ABC, calculamos os pontos médios das arestas: $D = (A+B)/2$, $E = (B+C)/2$, $F = (A+C)/2$. Cada ponto é então **projetado na esfera unitária** por normalização ($D \leftarrow D/\|D\|$). Isso substitui cada triângulo por 4 subtriângulos (ADF, DBE, FEC, DEF), multiplicando o número de faces por 4 a cada nível.

2.2 Unificação de Vértices (Tabela Hash)

Triângulos adjacentes compartilham arestas. Ao subdividir, o ponto médio de uma aresta compartilhada seria calculado duas vezes. Para unificar esses vértices, usamos uma **tabela hash** com coordenadas arredondadas como chave: $chave(v) = (round(x, 8), round(y, 8), round(z, 8))$. Se a chave já existe, retornamos o índice existente. Isso garante zero duplicatas e a validade da fórmula de Euler ($V - E + F = 2$).

2.3 Resultados — Evolução do Icosaedro

O icosaedro (12 vértices, 20 faces) é o sólido platônico que produz a esfera geodésica mais uniforme, por ter faces triangulares mais regulares. Após 3 níveis: 642 vértices e 1280 faces.

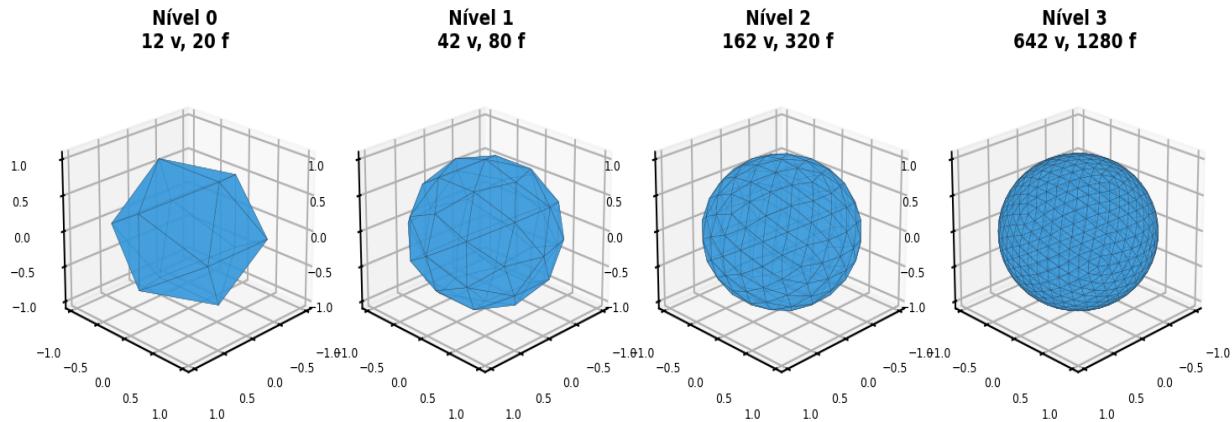


Figura 5: Evolução da esfera geodésica a partir do icosaedro — níveis 0 a 3.

2.4 Comparação dos Sólidos Iniciais

Com 3 níveis de refinamento para cada sólido platônico:

Tetraedro (4→256 faces): razão de áreas = 6.52 — triângulos muito desiguais.

Octaedro (8→512 faces): razão = 2.09 — melhor uniformidade.

Icosaedro (20→1280 faces): razão = 1.29 — quase perfeitamente uniforme.

A fórmula de Euler $V-E+F=2$ é satisfeita em todos os casos, confirmando a consistência topológica da malha.

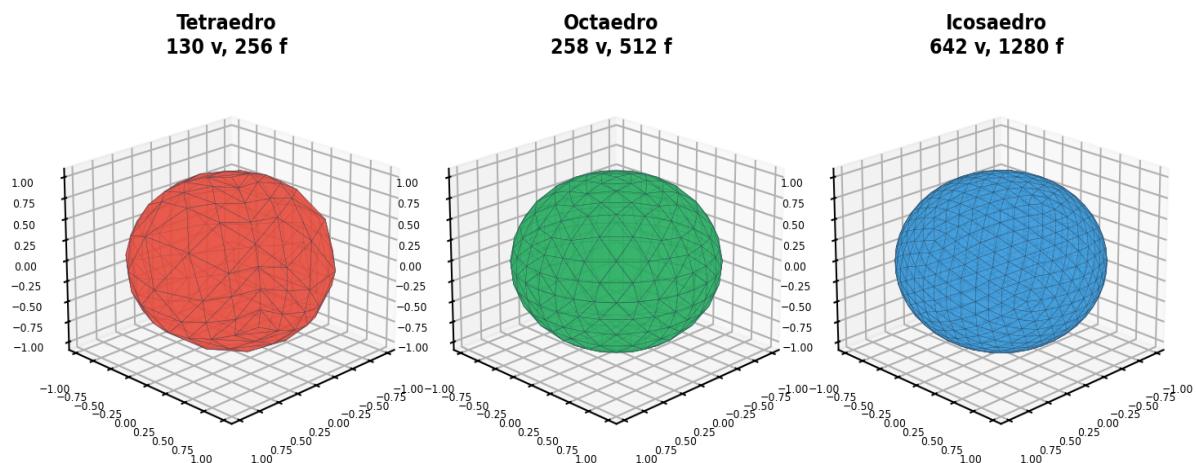


Figura 6: Comparação dos 3 sólidos platônicos após 3 subdivisões.

2.5 Verificação de Qualidade

Para cada esfera geodésica gerada, verificamos:

- **Norma dos vértices:** todos têm $\|v\| = 1$ com erro máximo de $\sim 10^{-16}$ (precisão de ponto flutuante).
- **Fórmula de Euler:** $V - E + F = 2$ para todos os casos.
- **Área total:** aproxima 4π com erro < 0.5%.
- **Uniformidade:** o icosaedro tem a menor razão entre áreas (1.29), confirmando por que é o preferido na prática (domos geodésicos de Buckminster Fuller).

2.6 Formato OBJ

Os modelos são exportados no formato Wavefront OBJ, com linhas $v \ x \ y \ z$ para vértices e $f \ i \ j \ k$ para faces (índices 1-based). Os arquivos gerados (`esfera_icosaedro.obj`, `esfera_octaedro.obj`, `esfera_tetraedro.obj`) podem ser abertos em qualquer visualizador 3D (MeshLab, Blender, etc.).

2.7 Dificuldades

A dificuldade mais sutil foi a unificação de vértices. Sem a tabela hash, vértices duplicados geram costuras na malha e violam a fórmula de Euler. Usar coordenadas arredondadas como chave resolve o problema de igualdade numérica em ponto flutuante. Outra questão foi manter a orientação consistente dos triângulos após a subdivisão, essencial para que as normais apontem todas para fora.

3. Conclusão

As duas tarefas exploram representações complementares em modelagem geométrica: a Tarefa 1 usa **representação implícita** com enumeração espacial (uniforme e adaptativa via Quadtree), demonstrando como o TVI permite classificar células eficientemente; a Tarefa 2 usa **representação por fronteira** (malha triangular), com subdivisão e projeção para aproximar a esfera. A implementação em Python permitiu iterar rapidamente e gerar visualizações claras dos conceitos estudados em aula.

Referências

- Finkel, R. A. e Bentley, J. L. (1974). Quad Trees: A Data Structure for Retrieval on Composite Keys. *Acta Informatica*, 4(1).
- Suffern, K. G. (1990). Quadtree algorithms for contouring functions of two variables. *The Computer Journal*, 33(5).
- Figueiredo, L. H. Slides do curso Modelagem Geométrica, IMPA 2026.