13.472J/1.128J/2.158J/16.940J COMPUTATIONAL GEOMETRY

Lecture 19

Prof. N. M. Patrikalakis

Contents

Decomposition models

19.1 Introduction

Decomposition models are representations of solids via combinations (unions) of basic special building blocks glued together. Alternatively, decomposition models may be considered to represent solids in terms of a subdivision of space (see also Lecture 1 for more details on the classification of these models). Various types of decomposition models are created by:

- various building blocks
- various combination methods used to create the model.

In order of increasing complexity, decomposition models are classified as follows:

1. Exhaustive enumeration
2. Space subdivision
3. Cell decomposition

19.2 Exhaustive enumeration

19.2.1 Definition and construction methods

Exhaustive enumeration is a representation by means of nonoverlapping cubes of uniform size and orientation, see Figure 19.1. An object is represented by a three dimensional Boolean array. Each cell represents a cubic volume of space. If a cell intersects with the region of interest it has a true value. Otherwise, the value is false. This can be pictured as a box divided into 3D cubical pixels, with 0 assigned if empty and 1 assigned if full. This representation involves:

- A regular subdivision of 3D space within a cube of given size which is partitioned and oriented in a certain way within a global coordinate system. The subdivision is made up of sub-cubes (3D pixels) of given size. Reference and access to each sub-cube is made by three integer indices i, j, k.
- For fixed space of interest we need a 3-D array, C_{ijk} of binary data:

$$C_{ijk} = \begin{cases} 1 & \text{if the sub-cube } i, j, k \text{ intersects the solid} \\ 0 & \text{if the sub-cube } i, j, k \text{ is empty} \end{cases} \tag{19.1}$$

Construction of exhaustive enumeration models requires an alternate representation or measurements (eg. digital tomograghy, medical scanning, sonar data, acoustic tomography data, etc). Usually the primary data type for such construction is a B-Rep or a CSG model or another exhaustive enumeration model at different resolution, and cube location and orientation.

Operations on exhaustive enumeration models are easy. Boolean operations for example (especially for models within the same cube at the same resolution) are direct. Similarly visualization and integral computations are very easy. However, for higher quality rendering, filtering methods to estimate accurate surface normals may be involved [16].

The binary matrix (19.1) typically represents a valid solid. However disconnected cells or cells with low degree of connectivity as in Figure 19.2 are undesirable. For the results of Boolean operations, filtering may be needed to maintain connectivity of cells. Strict connectivity occurs when each full cell has at least one full neighbor across a face.

19.2.2 Applications

Applications of exhaustive enumeration methods include:

- Underwater environment representation.
- Finite element meshing (first step in an algorithm to build such a mesh).
- Medical 3D data representation.
- Preprocessing representation for speeding up operations on other representations (eg. approximating integral properties such as volume, center of gravity, moments of inertia).

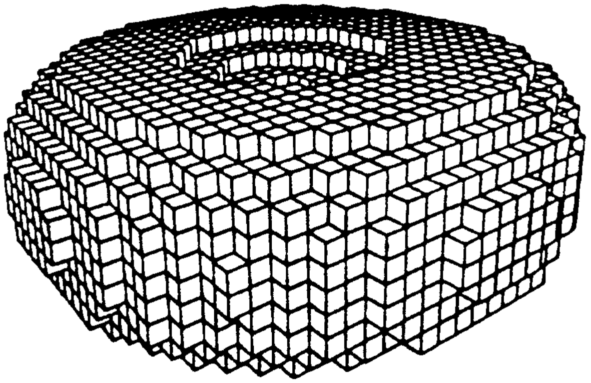


Figure 19.1: Exhaustive enumeration.

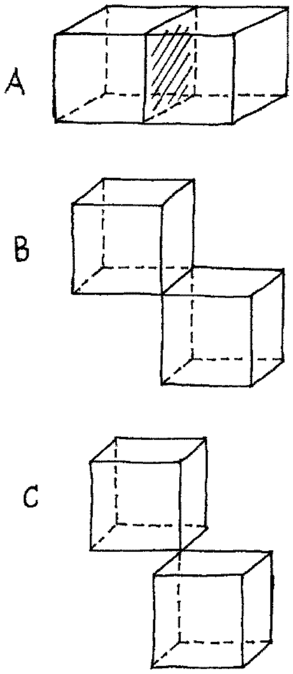


Figure 19.2: Various connectivities of cells in exhaustive enumeration models

19.2.3 Properties of exhaustive enumeration methods

Properties of exhaustive enumeration methods include:

- *Expressive power*: these methods are an approximation scheme and do not form a primary representation, especially within CAD/CAM applications.
- *Unambiguity and uniqueness* for fixed space (size, location and orientation of primary cube) and resolution. There do not exist different representations for the same object under these conditions (which is not true of many other representation methods such as B-rep or CSG described before).
- *Memory intensive*: eg. for a linear resolution of 256, 256^3 integer elements for C_{ijk} in equation 19.1 leading to $16M$ bits and this is a bare minimum.
- *Closure* [1] of operations (eg. Booleans).
- *Computational ease for algorithms*: VLSI implementation for volume rendering is possible. However, for high resolution the algorithm slows down.

[1] Closure means that an operation such as Boolean results in an object that can be represented by the same type of data structure.

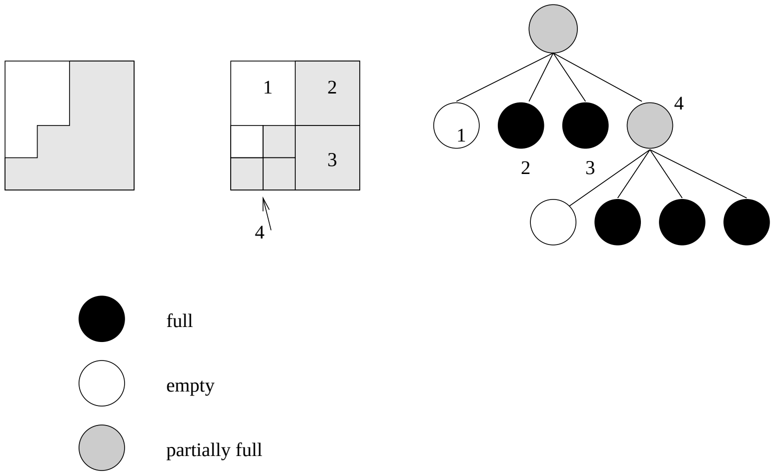


Figure 19.3: Quadtree representation.

19.3 Space subdivision

19.3.1 Motivation and definitions

Some of the motivations behind space subdivision methods include:

- Exhaustive enumeration is memory intensive and typically has low accuracy.
- Smaller memory requirements are possible, if adaptive subdivision is used;
- Octree/quadtree representations lead to a recursive subdivision into 8 octants (or 4 quadrants) that can be represented as an 8-ary tree (or 4-ary tree) for which efficient algorithms are also known.

In an octree representation a solid region is represented by hierarchically decomposing a usually cubic volume of space into successively smaller cubes (8 of them). Hierarchical division and cube orientation usually follows the spatial coordinate system. An example of quadtree, the two dimensional analogue, is shown Figure 19.3.

This is a trivial example. The method can continue to many more levels for a much more complex model. Some tolerance for the minimum size block is required. In addition, this very concise representation would become very large if the coordinate system was changed; for example, rotated 45 degrees.

This method leads to a quick way to compute the area and other integral properties of a region. It is often used in data analysis in fields such as medical applications and sonar imaging.

19.3.2 Construction of octrees

To create an octree, we apply a classification procedure to a given solid (represented using the Boundary Representation, Constructive Solid Geometry, or Exhaustive Enumeration methods, etc.) and decide if a given node of the octree is:

- Exterior to solid (white);
- Interior to solid (black);
- Partially interior to solid (grey).

The classification procedure is used recursively. It is based on Boolean solid operations, especially intersection. Figure 19.4 provides an simple example of octree representation.

In general, the decision if a given node of the octree is white, black, or grey is not an easy task. For the simple case of a convex solid object, it is sufficient to classify the eight vertices of the given node of the octree (which is a cube) with respect to the solid. This can be accomplished by for example casting a half-infinite ray from the point intersecting the solid's surfaces in a number of (multiplicity one) intersection points. If the number of such intersection points is even/odd, the point is outside/in (or on the surface of the solid). However, for a concave solid object, classification of the six faces of the cube with respect to the solid is necessary, see Figure 19.5 for an illustration in the 2-D case. This requires surface intersections with a planar patch. The memory and processing computation required for a 3-D object is on the order of the surface area of the object [12] [9]. Depending on the object and the resolution, this can still represent a large storage requirement.

19.3.3 Algorithms for octrees

Various algorithms for octrees are developed in Meagher [12] and are summarized here:

1. Tree generation or conversion from other representation methods were discussed above in Section 19.3.2.
2. Set operators (union, intersection, difference): A low resolution tree could be an effective preprocessor for a B-rep model in processes like interference checking.
3. Geometric transformations (translation, rotation, scaling).
4. Analysis procedures (integral, volume properties, connected components).
5. Rendering [16].

As an example we consider set or Boolean operations. Set operations lead to simple tree traversal

$$\text{Intersection(Tree A, Tree B)} = \text{Tree C}$$

Trees are traversed in synchronous fashion and a case analysis for the types of nodes is performed. We use the terms "black"= in-solid,"white" = out-of-solid. At each level of subdivision there are three cases [11]:

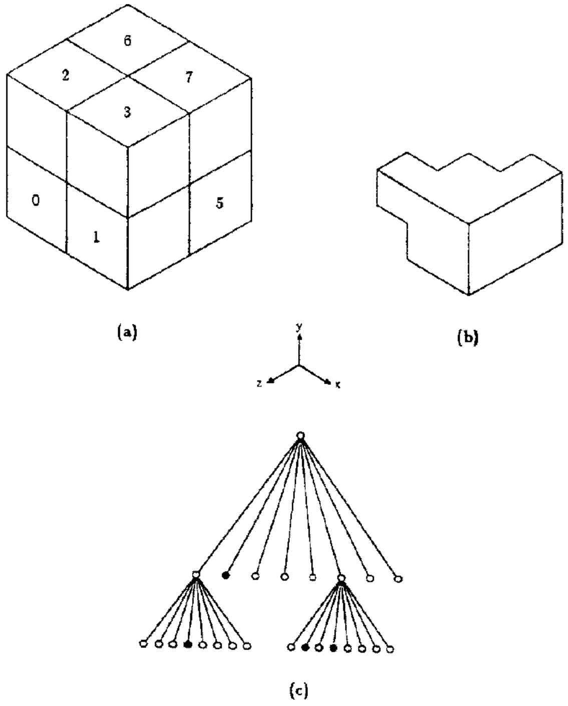


Figure 19.4: An octree model.

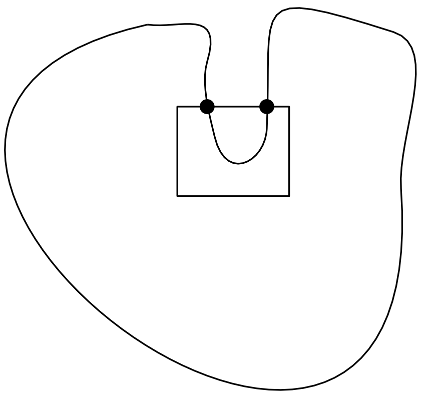


Figure 19.5: Classification of an quadtree node with respect to a concave object

1. If nodes n_1 and n_2 are both leaves, then the resulting node n_3 is black if n_1 and n_2 are both black; otherwise n_3 is white.

2. Either n_1 or n_2 is a leaf. If the leaf node is black, then the subtree of the non–leaf node is copied to the resultant octree; otherwise the resulting node is white.

3. If nodes n_1 and n_2 are non-leaves, then the algoritm considers recursively their children as above.

The complexity of such an intersection algorithm is proportional to the size of the smaller tree.

Not all algorithms are in the form of a simple tree traversal. Some algorithms may require at worst, a traversal up to the root and back down to a neighbor. Examples of such algorithms are surface rendering (shading), transparency rendering, and extraction of connected components.

19.3.4 Properties of octrees

Some of the properties of octrees include:

- *Expressive power*: they are an approximation scheme and do not form a primary representation.
- *Validity*: if no special connectivity is required, all octrees are valid.
- *Unambiguity and uniqueness*: for a fixed resolution there is only one compacted[2] octree;
- *Memory requirements*: not as large as exhaustive enumeration models, yet typically much larger than Boundary Representation and Constructive Solid Geometry models;
- *Closure of operations*: for example for Boolean operations and transformations;
- *Computational ease*: octrees are somewhat more complex than exhaustive enumeration.

[2] Algorithms such as set operations can create octrees with unnecessary nodes (eg. an internal nodes whose children are all black). Such nodes can be removed with a relatively simple tree traversal algorithm.

19.3.5 Binary space subdivision

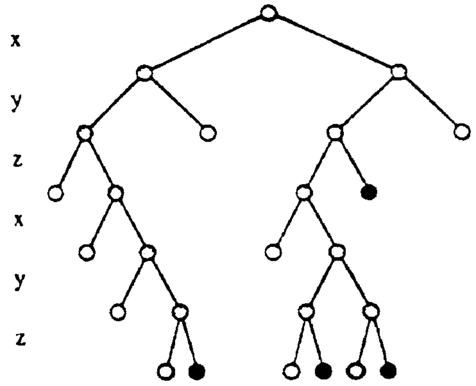


Figure 19.6: Binary subdivision tree.

Beyond octrees, an alternative type of tree representation involves dividing nodes into 2 rather than 8 components, see Mäntylä [11] and Figure 19.6. Subdivisions are performed in the X, Y, and Z coordinate directions sequentially. Binary trees are typically somewhat smaller than octrees and they can be converted to linear arrays containing special symbols [11].

19.4 Cell decompositions

19.4.1 Motivation

The motivation for cell decomposition methods is:

- Use of elements other than cubes, see Figure 19.7 for an example.
- Applications such as design of inhomogeneous (eg. composites) and functionally graded materials, finite element analysis methods, scientific visualization of scalar and vector fields.
- Cells are parametrized instances of a generic cell type, eg. a cell bounded by quadratic curves and surfaces.
- Cells are homeomorphic to spheres.
- Cells meet at a vertex, edge, face otherwise the representation is invalid.
- Cells are disjoint and non-overlapping.
- Cells may belong to different cell types, eg. box-like, tetrahedra-like, etc.

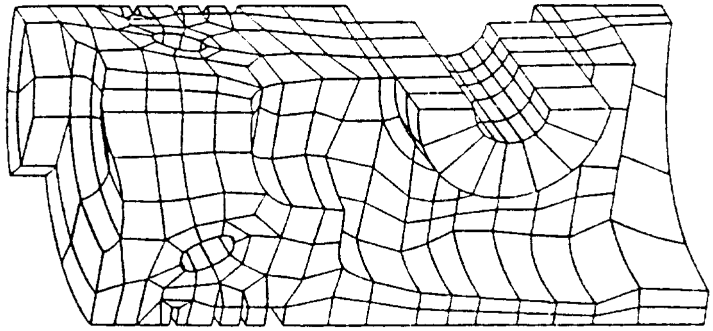


Figure 19.7: A cell decomposition (finite element mesh).

19.4.2 Cell tuple data structure

A cell decomposition can be represented using the *cell-tuple data structure* [2] which applies for n-D models, see also [1] for a review and summary of other related data structures such as the Quad-edge structure [6] for 2D models and the Facet-edge pair structure [4, 5] for 3-D models. Figures 19.8 and 19.9 present 2D and 3D examples. This data structure can be mapped into a relational database or a graph structure.

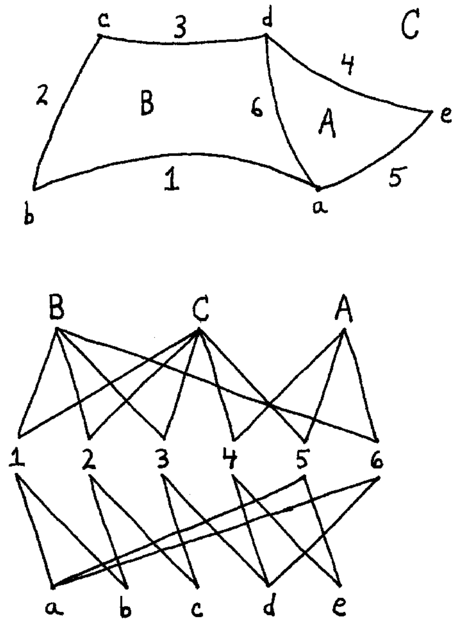


Figure 19.8: Cell data structure for a 2D model

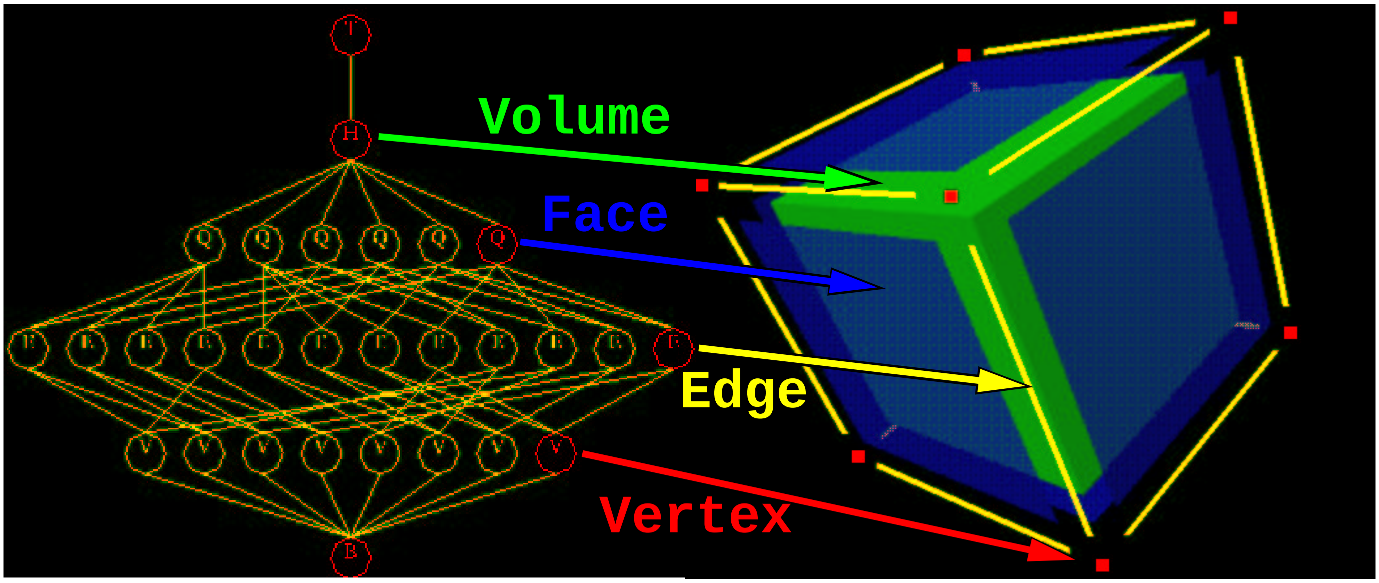


Figure 19.9: Cell data structure for a 3D model

19.4.3 Properties of cell decompositions

The properties of cell decomposition methods are:

- *Expressive power*: they are very general and accurate, not necessarily requiring approximations;
- *Validity*: they require an intersection test for verification;
- *Unambiguity*: they provide an unambiguous representation;
- *Nonuniqueness*: Similarly to the Constructive Solid Geometry method, the same object can be represented at different resolutions or with different types of mesh (eg. hexahedral, tetrahedral, etc.);
- *Generation*: It typically is done by conversion from other representations with or without geometric approximation;
- *Conciseness*: memory utilization is less than octrees, yet more than Boundary Representation;
- *Applicability*: finite element meshing, multimaterial non-homogeneous objects, visualization of fields, etc.

Integral properties of geometric models

19.5 Introduction

One of the important advantages of using a CAD model for representing and designing an object is that we can easily compute the integral properties of such models such as edge curves, faces and volumes. Integral properties include length, area, centroid, moment of inertia, and volume. These are very useful in preliminary design. For example, surface area affects drag, volume affects the carrying capacity of a vehicle, centroids are useful in hydrostatic balance, moments of inertia are used in dynamics and in hydrostatic stability calculation (for ships).

Computation of the integral properties of curves, surface patches and solids involves evaluation of single, double and triple integrals of the form

$$\phi_{curve} = \int_{curve} f(\mathbf{P})dL, \qquad \phi_{surface} = \int_{surface} f(\mathbf{P})dS, \qquad \phi_{solid} = \int_{solid} f(\mathbf{P})dV \qquad (19.2)$$

where ϕ is the required property, $\mathbf{P}$ is a point and f is a real-valued function, which depends on the type of property required. We have studied three classes of solid representation methods in the previous chapters, namely Decomposition methods, Constructive Solid Geometry (CSG) methods, and Boundary Representation (B-rep) methods.

For decomposition methods, the integral over the solid reduces to a sum of integrals

$$\int_{solid} f dV = \sum_i \int_{cell_i} f dV \qquad (19.3)$$

where $cell_i$ is the $i - th$ cell which is either full or partially full. For the case of exhaustive spatial enumeration the cells are constant-sized cubes and for the octree decompositions they are variable-sized cubes [10, 13]. For high resolution models it is enough to consider all the $cell_i$ to be full entirely and for those cases the resulting integrals are elementary and can be computed using simple analytic forms.

As we have studied in the previous chapters, CSG is a tree whose nodes represent the Boolean operators and the leaves are the primitive solids. Therefore the computation of integral properties of CSG solids consists of applying the following formula recursively [10]:

$$\begin{aligned} \int_{A\cup B} f dV &= \int_A f dV + \int_B f dV - \int_{A\cap B} f dV & (19.4) \\ \int_{A-B} f dV &= \int_A f dV - \int_{A\cap B} f dV. & (19.5) \end{aligned}$$

Consequently we need to compute integrals over primitives (which can be evaluated analytically) and integrals involving intersections of primitives $\int_{A\cap B} f dV$ which can be approximated using a ray casting, ray classification and integral approximation method.

Boundary representation, which is the most generally used representation today, represents the object in terms of their boundary elements (e.g. vertices, edges, faces). For evaluating the integral properties for B-rep solids, the following theorems from vector calculus are useful [7]:

1. **Green's Theorem**
   If C is a piecewise smooth, simple closed curve that bounds a region R, and if $P(x, y)$ and $Q(x, y)$ are continuous functions which have continuous partial derivatives along C and throughout R, then

$$\oint_C (Pdx + Qdy) = \iint_R \left(\frac{\partial Q}{\partial x} - \frac{\partial P}{\partial y} \right) dA \tag{19.6}$$

2. **Divergence Theorem** (also called *Gauss' Theorem*)
   The flux of vector field $\mathbf{F}$ flowing outward through a closed surface S equals the integral of the divergence of $\mathbf{F}$ over the region R bounded by S;

$$\iint_S \mathbf{F} \cdot \mathbf{n} dA = \iiint_R \nabla \mathbf{F} dV \tag{19.7}$$

   where $\mathbf{n}$ is the outward unit normal vector and

$$\nabla \cdot \mathbf{F} = \frac{\partial \mathbf{F}}{\partial x} \cdot \mathbf{i} + \frac{\partial \mathbf{F}}{\partial y} \cdot \mathbf{j} + \frac{\partial \mathbf{F}}{\partial z} \cdot \mathbf{k}. \tag{19.8}$$

   where $\mathbf{i}$, $\mathbf{j}$, $\mathbf{k}$ are the unit coordinate vectors.

In the sequel we will apply these theorems to compute the integral properties of geometric models represented by the B-rep method (in 1-3 dimensions).

19.6 Integral properties of curves

19.6.1 Planar curves

Let a planar curve be defined by

$$\mathbf{r} = (x(t), y(t))\,, \quad t_o \le t \le t_1 \tag{19.9}$$

- Length

$$\begin{aligned} L &= \int_{s_0}^{s_1} ds = \int_{t_0}^{t_1} \sqrt{\dot{\mathbf{r}}(t) \cdot \dot{\mathbf{r}}(t)} dt \\ &= \int_{t_0}^{t_1} \sqrt{\dot{x}^2(t) + \dot{y}^2(t)} dt \end{aligned} \tag{19.10}$$

- “Centroid”

$$\begin{aligned}\mathbf{r}_c &= (x_c, y_c) = \frac{\int_{s_0}^{s_1} \mathbf{r} ds}{\int_{s_0}^{s_1} ds} \qquad (19.11)\\ &= \frac{1}{L}\int_{t_0}^{t_1} \mathbf{r}(t)\sqrt{\dot{x}^2(t)+\dot{y}^2(t)}dt\end{aligned}$$

- “Moments of inertia”

$$\begin{aligned}I_{xx} &= \int_{s_0}^{s_1} y^2 ds = \int_{t_0}^{t_1} y^2(t)\sqrt{\dot{x}^2(t)+\dot{y}^2(t)}dt \qquad (19.12)\\ I_{yy} &= \int_{s_0}^{s_1} x^2 ds = \int_{t_0}^{t_1} x^2(t)\sqrt{\dot{x}^2(t)+\dot{y}^2(t)}dt \qquad (19.13)\\ I_{xy} &= \int_{s_0}^{s_1} xy ds = \int_{t_0}^{t_1} x(t)y(t)\sqrt{\dot{x}^2(t)+\dot{y}^2(t)}dt \qquad (19.14)\end{aligned}$$

19.6.2 3D curves

Let a 3D curve be defined by

$$\mathbf{r} = (x(t), y(t), z(t)), \quad t_o \le t \le t_1 \qquad (19.15)$$

- Length

$$\begin{aligned}L &= \int_{s_0}^{s_1} ds = \int_{t_0}^{t_1} \sqrt{\dot{\mathbf{r}}(t)\cdot\dot{\mathbf{r}}(t)}dt \qquad (19.16)\\ &= \int_{t_0}^{t_1} \sqrt{\dot{x}^2(t)+\dot{y}^2(t)+\dot{z}^2(t)}dt\end{aligned}$$

- “Centroid”

$$\begin{aligned}\mathbf{r}_c &= (x_c, y_c, z_c) = \frac{\int_{s_0}^{s_1} \mathbf{r} ds}{\int_{s_0}^{s_1} ds} \qquad (19.17)\\ &= \frac{1}{L}\int_{t_0}^{t_1} \mathbf{r}(t)\sqrt{\dot{x}^2(t)+\dot{y}^2(t)+\dot{z}^2(t)}dt\end{aligned}$$

- “Moments of inertia”

$$\begin{aligned}I_{xx} &= \int_{s_0}^{s_1} (y^2+z^2) ds = \int_{t_0}^{t_1} \left(y^2(t)+z^2(t)\right)\sqrt{\dot{x}^2(t)+\dot{y}^2(t)+\dot{z}^2(t)}dt \qquad (19.18)\\ I_{yy} &= \int_{s_0}^{s_1} (x^2+z^2) ds = \int_{t_0}^{t_1} \left(x^2(t)+z^2(t)\right)\sqrt{\dot{x}^2(t)+\dot{y}^2(t)+\dot{z}^2(t)}dt \qquad (19.19)\\ I_{zz} &= \int_{s_0}^{s_1} (x^2+y^2) ds = \int_{t_0}^{t_1} \left(x^2(t)+y^2(t)\right)\sqrt{\dot{x}^2(t)+\dot{y}^2(t)+\dot{z}^2(t)}dt \qquad (19.20)\\ I_{xy} &= \int_{s_0}^{s_1} xy ds = \int_{t_0}^{t_1} x(t)y(t)\sqrt{\dot{x}^2(t)+\dot{y}^2(t)+\dot{z}^2(t)}dt \qquad (19.21)\\ I_{yz} &= \int_{s_0}^{s_1} yz ds = \int_{t_0}^{t_1} y(t)z(t)\sqrt{\dot{x}^2(t)+\dot{y}^2(t)+\dot{z}^2(t)}dt \qquad (19.22)\\ I_{xz} &= \int_{s_0}^{s_1} xz ds = \int_{t_0}^{t_1} x(t)z(t)\sqrt{\dot{x}^2(t)+\dot{y}^2(t)+\dot{z}^2(t)}dt \qquad (19.23)\end{aligned}$$

19.7 Integral properties of surface patches

19.7.1 Planar regions

Let us consider a planar region as in Figure 19.10

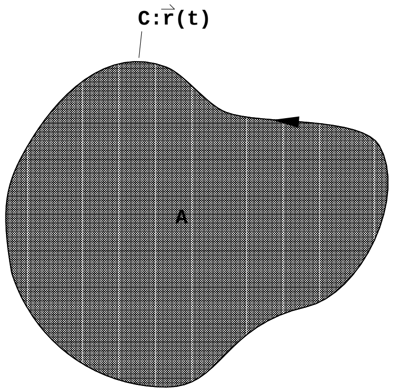


Figure 19.10: Planar region A

- Area

$$A = \iint_A dA \tag{19.24}$$

Using Green's theorem with $Q(x, y) = x$ and $P(x, y) = -y$, then

$$\frac{1}{2}(Q_x - P_y) = \frac{1}{2}(1 + 1) = 1 \tag{19.25}$$

where subscripts x, y denote partial derivatives. We can rewrite equation (19.24) using *Green's theorem* as

$$\begin{aligned} A &= \iint_A \frac{1}{2}(Q_x - P_y) dx dy \\ &= \frac{1}{2}\oint_C (P dx + Q dy) \\ &= \frac{1}{2}\oint_C (-y dx + x dy) \\ &= \frac{1}{2}\oint_C (x\dot{y} - y\dot{x}) dt \end{aligned} \tag{19.26}$$

If $x(t), y(t)$ are piecewise polynomial functions, the above integral can be evaluated from a symbolic/analytic integration formula but this is typically tedious. By contrast, numerical integration methods may be used more easily (see Appendix).

- Centroid

$$\mathbf{r}_c = (x_c, y_c) = \frac{\iint_A \mathbf{r} dA}{\iint_A dA},$$

where A is the shaded area, and

$$\iint_A \mathbf{r} dA = \left(\iint_A x dA, \iint_A y dA \right). \tag{19.27}$$

Let $Q(x,y) = \frac{x^2}{2}, P(x,y) = 0$, then

$$Q_x - P_y = x - 0 = x \tag{19.28}$$

Therefore

$$\iint_A x dA = \oint_C P dx + Q dy = \oint_C x^2 dy \tag{19.29}$$

where C is the complete boundary of A. Thus,

$$\begin{aligned} x_c &= \frac{\iint_A x dA}{\iint_A dA} \\ &= \frac{1}{A} \iint_A x dA \\ &= \frac{1}{A} \oint_C x^2 \dot{y} dt \end{aligned} \tag{19.30}$$

Let $Q(x,y) = 0, P(x,y) = -\frac{y^2}{2}$, then

$$Q_x - P_y = 0 + y = y \tag{19.31}$$

Similarly,

$$\begin{aligned} y_c &= \frac{\iint_A y dA}{\iint_A dA} \\ &= \frac{1}{A} \iint_A y dA \\ &= -\frac{1}{A} \oint_C y^2 \dot{x} dt \end{aligned} \tag{19.32}$$

- Moments of inertia

  1. $I_{xx} = \iint_A y^2 dA$
     Let $Q(x,y) = 0, P(x,y) = -\frac{y^3}{3}$, then

     $$Q_x - P_y = y^2 \tag{19.33}$$

     Using *Green's Theorem*,

     $$\begin{aligned} I_{xx} &= \iint_A (Q_x - P_y) dx dy = \oint_C -y^3 dx \\ &= -\oint_C y^3 \dot{x} dt \end{aligned} \tag{19.34}$$

2. $I_{yy} = \iint_A x^2 dA$
   Let $Q(x,y) = \frac{x^3}{3}, P(x,y) = 0$, then

$$Q_x - P_y = x^2 \tag{19.35}$$

Using *Green's Theorem*,

$$\begin{aligned} I_{yy} &= \iint_A (Q_x - P_y) dxdy = \frac{1}{4} \oint_C \frac{x^3}{3} dy \\ &= \oint_C x^3 \dot{y} dt \end{aligned} \tag{19.36}$$

3. $I_{xy} = \iint_A xy dA$
   Let $Q(x,y) = \frac{x^2 y}{2}, P(x,y) = 0$, then

$$Q_x - P_y = xy + 0 = xy \tag{19.37}$$

Using *Green's Theorem*,

$$\begin{aligned} I_{xy} &= \iint_A (Q_x - P_y) dxdy \\ &= \frac{1}{2} \oint_C x^2 y \dot{y} dt \end{aligned} \tag{19.38}$$

If $x(t), y(t)$ are piecewise polynomial functions, the above integrals can be evaluated from a symbolic/analytic integration formula but this is typically tedious. By contrast, numerical integration methods may be used more easily (see Appendix).

19.7.2 Curved surface patch

Let us consider a curved surface patch $\mathbf{r} = \mathbf{r}(u,v)$, with $(u,v) \in A$, where A is a given parametric domain, as in Figure 19.11.

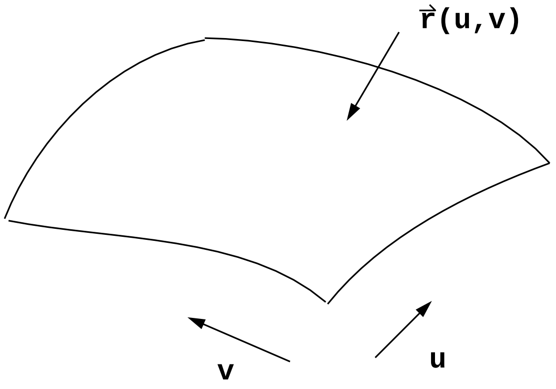


Figure 19.11: Curved surface patch

- Area

$$\begin{aligned} A &= \iint_A dA \\ &= \iint_A |\mathbf{r}_u \times \mathbf{r}_v| du dv = \iint_A \sqrt{EG - F^2} du dv \end{aligned} \tag{19.39}$$

where E, F and G are the first fundamental form coefficients $E = \mathbf{r}_u \cdot \mathbf{r}_u, \quad F = \mathbf{r}_u \cdot \mathbf{r}_v, \quad G = \mathbf{r}_v \cdot \mathbf{r}_v$. (see Chapter 2).

- “Centroid”

$$\begin{aligned} \mathbf{r}_c &= (x_c, y_c, z_c) = \frac{\iint_A \mathbf{r} dA}{\iint_A dA} \\ &= \frac{1}{A} \iint_A [x(u,v), y(u,v), z(u,v)] \sqrt{EG - F^2} du dv \end{aligned} \tag{19.40}$$

- “Moments of inertia”

$$I_{xx} = \iint_A [y^2(u,v) + z^2(u,v)] \sqrt{EG - F^2} du dv \tag{19.41}$$

$$I_{yy} = \iint_A [x^2(u,v) + z^2(u,v)] \sqrt{EG - F^2} du dv \tag{19.42}$$

$$I_{zz} = \iint_A [x^2(u,v) + y^2(u,v)] \sqrt{EG - F^2} du dv \tag{19.43}$$

$$I_{xy} = \iint_A [x(u,v) y(u,v)] \sqrt{EG - F^2} du dv \tag{19.44}$$

$$I_{xz} = \iint_A [x(u,v) z(u,v)] \sqrt{EG - F^2} du dv \tag{19.45}$$

$$I_{yz} = \iint_A [y(u,v) z(u,v)] \sqrt{EG - F^2} du dv \tag{19.46}$$

Integrals 19.39-19.46 may be evaluated numerically as in the Appendix.

19.8 Solids

For solids described by the B-rep method it is convenient to transform volume integrals into surface integrals by means of the divergence theorem.

- Volume

$$V = \iiint_V dV \tag{19.47}$$

Choose

$$\mathbf{r} = x\mathbf{i} + y\mathbf{j} + z\mathbf{k} \tag{19.48}$$

then

$$\nabla \cdot \mathbf{r} = \frac{\partial \mathbf{r}}{\partial x} \cdot \mathbf{i} + \frac{\partial \mathbf{r}}{\partial y} \cdot \mathbf{j} + \frac{\partial \mathbf{r}}{\partial z} \cdot \mathbf{k} = 3 \tag{19.49}$$

Using the *Divergence (or Gauss') Theorem*,

$$\begin{aligned} V &= \iiint_V dV = \frac{1}{3}\iiint_V \nabla \cdot \mathbf{r} dV \\ &= \frac{1}{3}\iint_A \mathbf{r}\cdot\mathbf{n} dA = \frac{1}{3}\iint_A \mathbf{r}\cdot\mathbf{n}|\mathbf{r}_u \times \mathbf{r}_v| dudv \\ &= \frac{1}{3}\iint_A \mathbf{r}\cdot(\mathbf{r}_u \times \mathbf{r}_v) dudv \qquad (19.50) \end{aligned}$$

given that

$$\mathbf{n} = \frac{\mathbf{r}_u \times \mathbf{r}_v}{|\mathbf{r}_u \times \mathbf{r}_v|} \qquad (19.51)$$

- Centroid

$$\mathbf{r}_c = (x_c, y_c, z_c) = \frac{\iiint_V \mathbf{r} dV}{\iiint_V dV} \qquad (19.52)$$

Choose

$$\mathbf{r} = \frac{1}{2}x^2\mathbf{i} \qquad (19.53)$$

then

$$\begin{aligned} \nabla \cdot \mathbf{r} &= \frac{\partial \mathbf{r}}{\partial x}\cdot\mathbf{i} + \frac{\partial \mathbf{r}}{\partial y}\cdot\mathbf{j} + \frac{\partial \mathbf{r}}{\partial z}\cdot\mathbf{k} = x \qquad (19.54) \\ x_c &= \frac{1}{V}\iiint_V x dV = \frac{1}{V}\iiint_V \nabla\cdot\mathbf{r} dV \\ &= \frac{1}{V}\iint_A \frac{1}{2}x^2(\mathbf{i}\cdot\mathbf{n}) dA \\ &= \frac{1}{V}\iint_A \frac{1}{2}x^2(\mathbf{i}\cdot(\mathbf{r}_u \times \mathbf{r}_v)) dudv \qquad (19.55) \end{aligned}$$

Similarly, expressions are obtained for y_c, z_c:

$$\begin{aligned} y_c &= \frac{1}{V}\iint_A \frac{1}{2}y^2(\mathbf{j}\cdot(\mathbf{r}_u \times \mathbf{r}_v)) dudv \qquad (19.56) \\ z_c &= \frac{1}{V}\iint_A \frac{1}{2}z^2(\mathbf{k}\cdot(\mathbf{r}_u \times \mathbf{r}_v)) dudv \qquad (19.57) \end{aligned}$$

- Moments of inertia

$$I_{xx} = \iiint_V (y^2 + z^2) dV \qquad (19.58)$$

Choose

$$\mathbf{r} = (y^2 + z^2)x\mathbf{i} \qquad (19.59)$$

then

$$\nabla \cdot \mathbf{r} \;=\; \frac{\partial \mathbf{r}}{\partial x}\cdot \mathbf{i} + \frac{\partial \mathbf{r}}{\partial y}\cdot \mathbf{j} + \frac{\partial \mathbf{r}}{\partial z}\cdot \mathbf{k} = y^2 + z^2 \tag{19.60}$$

Thus

$$\begin{aligned} I_{xx} &= \iiint_V (y^2+z^2)dV = \iiint_V \nabla\cdot\mathbf{r}dV \\ &= \iint_A (y^2+z^2)x(\mathbf{i}\cdot\mathbf{n})dA \\ &= \iint_A (y^2+z^2)x(\mathbf{i}\cdot(\mathbf{r}_u\times\mathbf{r}_v))dudv \qquad (19.61) \end{aligned}$$

Similarly,

$$I_{yy} = \iint_A (x^2+z^2)y(\mathbf{j}\cdot(\mathbf{r}_u\times\mathbf{r}_v))dudv \tag{19.62}$$

$$I_{zz} = \iint_A (x^2+y^2)z(\mathbf{k}\cdot(\mathbf{r}_u\times\mathbf{r}_v))dudv \tag{19.63}$$

$$I_{xy} = \iint_A xyz(\mathbf{k}\cdot(\mathbf{r}_u\times\mathbf{r}_v))dudv \tag{19.64}$$

$$I_{xz} = \iint_A xzy(\mathbf{j}\cdot(\mathbf{r}_u\times\mathbf{r}_v))dudv \tag{19.65}$$

$$I_{yz} = \iint_A yzx(\mathbf{i}\cdot(\mathbf{r}_u\times\mathbf{r}_v))dudv \tag{19.66}$$

19.9 Example: solid of revolution

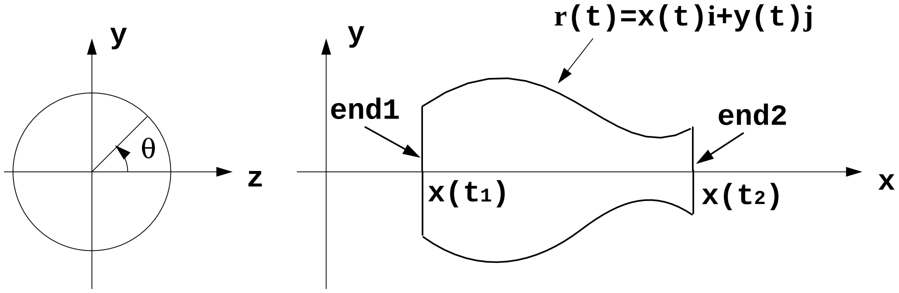


Figure 19.12: Solid of revolution

Let a solid of revolution be defined by

$$\begin{aligned} &\mathbf{r}(t,\theta) = x(t)\mathbf{i} + y(t)\sin\theta\mathbf{j} + y(t)\cos\theta\mathbf{k}, \qquad (19.67) \\ &t_1 \le t \le t_2, \quad 0 \le \theta \le 2\pi. \end{aligned}$$

and we assume that the two end caps are closed off with planar disks (see Figure 19.12).

- Surface area of surface of revolution (with end caps)

$$\begin{aligned} \mathbf{r}_t &= (\dot{x}, \dot{y}\sin\theta, \dot{y}\cos\theta) && (19.68) \\ \mathbf{r}_\theta &= (0, y\cos\theta, -y\sin\theta) && (19.69) \end{aligned}$$

$$\begin{aligned} \mathbf{r}_t \times \mathbf{r}_\theta &= \begin{vmatrix} \mathbf{i} & \mathbf{j} & \mathbf{k} \\ \dot{x} & \dot{y}\sin\theta & \dot{y}\cos\theta \\ 0 & y\cos\theta & -y\sin\theta \end{vmatrix} \\ &= (-y\dot{y}\sin^2\theta - y\dot{y}\cos^2\theta)\mathbf{i} + \dot{x}y\sin\theta\mathbf{j} + \dot{x}y\cos\theta\mathbf{k} \\ &= -y\dot{y}\mathbf{i} + \dot{x}y\sin\theta\mathbf{j} + \dot{x}y\cos\theta\mathbf{k} && (19.70) \end{aligned}$$

$$|\mathbf{r}_t \times \mathbf{r}_\theta| = \sqrt{y^2\dot{y}^2 + \dot{x}^2y^2} = y\sqrt{\dot{x}^2 + \dot{y}^2} \quad (19.71)$$

$$\begin{aligned} A &= \iint_A dA = \int_{t_1}^{t_2}\int_0^{2\pi} y\sqrt{\dot{x}^2+\dot{y}^2}d\theta dt \\ &= 2\pi\int_{t_1}^{t_2} y\sqrt{\dot{x}^2+\dot{y}^2}dt && (19.72) \end{aligned}$$

- Volume

$$\mathbf{r}_t \times \mathbf{r}_\theta = (-y\dot{y}, \dot{x}y\sin\theta, \dot{x}y\cos\theta) \quad (19.73)$$

$$\begin{aligned} \mathbf{r}\cdot(\mathbf{r}_t \times \mathbf{r}_\theta) &= (x, y\sin\theta, y\cos\theta)\cdot(-y\dot{y}, \dot{x}y\sin\theta, \dot{x}y\cos\theta) \\ &= -xy\dot{y} + \dot{x}y^2\sin^2\theta + \dot{x}y^2\cos^2\theta \\ &= -xy\dot{y} + \dot{x}y^2 && (19.74) \end{aligned}$$

$$\begin{aligned} V &= \frac{1}{3}\int_{t_1}^{t_2}\int_0^{2\pi}\mathbf{r}\cdot(\mathbf{r}_t\times\mathbf{r}_\theta)dtd\theta - \frac{1}{3}\iint_{A_{end1}} xdxdy + \frac{1}{3}\iint_{A_{end2}} xdxdy \\ &= \frac{1}{3}\int_{t_1}^{t_2}\int_0^{2\pi}(-xy\dot{y}+\dot{x}y^2)d\theta dt - \frac{\pi}{3}x(t_1)y^2(t_1) + \frac{\pi}{3}x(t_2)y^2(t_2) \\ &= \frac{2\pi}{3}\int_{t_1}^{t_2}(-xy\dot{y}+\dot{x}y^2)dt - \frac{\pi}{3}x(t_1)y^2(t_1) + \frac{\pi}{3}x(t_2)y^2(t_2) && (19.75) \end{aligned}$$

Using integration by parts,

$$\begin{aligned} -\int_{t_1}^{t_2} xy\dot{y}dt &= -[xy^2]_{t_1}^{t_2} + \int_{t_1}^{t_2}(xy)'ydt \\ &= -x(t_2)y^2(t_2) + x(t_1)y^2(t_1) + \int_{t_1}^{t_2}(\dot{x}y^2 + xy\dot{y})dt && (19.76) \end{aligned}$$

Thus

$$-2\int_{t_1}^{t_2} xy\dot{y}dt = -x(t_2)y^2(t_2) + x(t_1)y^2(t_1) + \int_{t_1}^{t_2}\dot{x}y^2dt \quad (19.77)$$

The volume, therefore, is

$$\begin{aligned} V &= \frac{2\pi}{3}\int_{t_1}^{t_2} \dot{x}y^2 dt - \frac{2\pi}{3}\int_{t_1}^{t_2}(-xy\dot{y})dt - \frac{\pi}{3}x(t_1)y^2(t_1) + \frac{\pi}{3}x(t_2)y^2(t_2) \\ &= \pi\int_{t_1}^{t_2} \dot{x}y^2 dt \end{aligned} \tag{19.78}$$

(corroborating the obvious formula from elementary calculus)

- Centroid

  1. $\mathbf{i}\cdot(\mathbf{r}_t \times \mathbf{r}_\theta) = -y\dot{y}$

  $$\begin{aligned} x_c &= \frac{1}{V}\int_{t_1}^{t_2}\int_0^{2\pi} \frac{1}{2}x^2(-y\dot{y})dtd\theta - \frac{\pi}{2V}x^2(t_1)y^2(t_1) + \frac{\pi}{2V}x^2(t_2)y^2(t_2) \\ &= \frac{\pi}{V}\int_{t_1}^{t_2}(-x^2y\dot{y})dt + \frac{\pi}{2V}(x^2(t_2)y^2(t_2) - x^2(t_1)y^2(t_1)) \end{aligned} \tag{19.79}$$

  Integrate by parts

  $$\begin{aligned} -\int_{t_1}^{t_2} x^2y\dot{y}dt &= -[x^2y^2]_{t_1}^{t_2} + \int_{t_1}^{t_2}(x^2y)'ydt \\ &= -x^2(t_2)y^2(t_2) + x^2(t_1)y^2(t_1) + \int_{t_1}^{t_2}(2x\dot{x}y + x^2\dot{y})ydt \end{aligned} \tag{19.80}$$

  Thus

  $$-2\int_{t_1}^{t_2} x^2y\dot{y}dt = -x^2(t_2)y^2(t_2) + x^2(t_1)y^2(t_1) + 2\int_{t_1}^{t_2} x\dot{x}y^2dt \tag{19.81}$$

  and

  $$x_c = \frac{\pi}{V}\int_{t_1}^{t_2} x\dot{x}y^2dt \tag{19.82}$$

  (corroborating the obvious formula from elementary calculus)

  2. $\mathbf{j}\cdot(\mathbf{r}_t \times \mathbf{r}_\theta) = \dot{x}y\sin\theta$

  $$\begin{aligned} y_c &= \frac{1}{V}\iint_A \frac{1}{2}y^2\dot{x}y\sin\theta dtd\theta \\ &= \frac{1}{2V}\int_{t_1}^{t_2}\int_0^{2\pi} y^3\dot{x}\sin\theta d\theta dt \\ &= \frac{1}{2V}\int_{t_1}^{t_2} y^3\dot{x}[-\cos\theta]_0^{2\pi}dt \\ &= 0 \end{aligned} \tag{19.83}$$

  3. $\mathbf{k}\cdot(\mathbf{r}_t \times \mathbf{r}_\theta) = \dot{x}y\cos\theta$

  $$\begin{aligned} z_c &= \frac{1}{V}\iint_A \frac{1}{2}z^2\dot{x}y\cos\theta dtd\theta \\ &= \frac{1}{2V}\int_{t_1}^{t_2}\int_0^{2\pi} \dot{x}yz^2\cos\theta d\theta dt \\ &= \frac{1}{2V}\int_{t_1}^{t_2} \dot{x}yz^2[\sin\theta]_0^{2\pi}dt \\ &= 0 \end{aligned} \tag{19.84}$$

19.10 Appendix: Review of numerical integration methods

19.10.1 Trapezoidal rule of integration

Here we compute $\int_a^b f(x)dx$ as in Figure 19.13 by decomposing the integral into n subintervals [3]. The area over each subinterval is approximated by the area of the trapezoid. The integral is then obtained by the sum of areas over all the subintervals.

- Area in each subinterval $[x_i, x_{i+1}]$

$$\int_{x_i}^{x_{i+1}} f(x)dx \approx (f(x_i) + f(x_{i+1}))\frac{\Delta x}{2} = \frac{h}{2}(f_i + f_{i+1}) \tag{19.85}$$

  where $\Delta x = x_{i+1} - x_i$.

- $[a, b]$ subdivided into n-subintervals

$$\int_a^b f(x)dx = \sum_{i=1}^{n} \frac{h}{2}(f_i + f_{i+1}) = \frac{h}{2}(f_1 + 2f_2 + 2f_3 + \cdots + 2f_n + f_{n+1}) \tag{19.86}$$

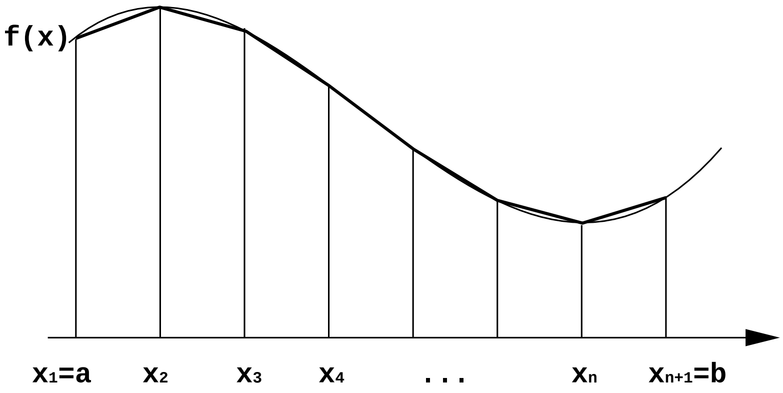


Figure 19.13: Trapezoidal rule of integration

- Error in *Trapezoidal rule* [3]

$$\begin{array}{ll} \text{Local error} & -\frac{h^3}{12}f''(\xi), \quad x_i < \xi < x_{i+1} \\ \text{Global error} & -\frac{h^2}{12}(b-a)f''(\xi), \quad a < \xi < b \end{array}$$

19.10.2 Simpson's rule of integration

For the trapezoidal rule, we approximate the curve by n straight line segments, while in *Simpson's rule* of integration we approximate it with a parabola in a piecewise manner [3].

$$y = ax^2 + bx + c. \tag{19.87}$$

Therefore, this rule requires the use of three base points.

- Area in two subintervals $[-h, h]$
  Let the three points be $x = -h, 0, h$, and the corresponding y values be f_1, f_2, and f_3, see Figure 19.14. Then,

$$\begin{aligned} f_1 &= ah^2 - bh + c \qquad (19.88)\\ f_2 &= c \qquad (19.89)\\ f_3 &= ah^2 + bh + c \qquad (19.90) \end{aligned}$$

  If we solve for a, b, and c, we obtain

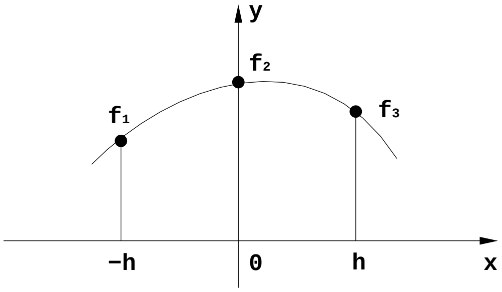


Figure 19.14: Simpson's rule

$$\begin{aligned} a &= \frac{f_1 - 2f_2 + f_3}{2h^2} \qquad (19.91)\\ b &= \frac{f_3 - f_1}{2h} \qquad (19.92)\\ c &= f_2 \qquad (19.93) \end{aligned}$$

  Thus,

$$\begin{aligned} \int_{-h}^{h} (ax^2 + bx + c)dx &= \frac{a}{3}[x^3]_{-h}^{h} + \frac{b}{2}[x^2]_{-h}^{h} + c[x]_{-h}^{h}\\ &= \frac{2a}{3}h^3 + 2ch\\ &= \frac{h}{3}[f_1 + 4f_2 + f_3] \qquad (19.94) \end{aligned}$$

- Area in $[a, b]$ subdivided into n-subintervals

$$\int_a^b f(x)dx = \frac{h}{3}(f_1 + 4f_2 + 2f_3 + 4f_4 + 2f_5 + \cdots + 2f_{n-1} + 4f_n + f_{n+1}) \tag{19.95}$$

- Error in Simpson's rule [3]

$$\begin{array}{ll} \text{Local error} & -\frac{h^5}{90}f^{(4)}(\xi), \quad x_i < \xi < x_{i+1} \\ \text{Global error} & -\frac{b-a}{180}h^4 f^{(4)}(\xi), \quad a < \xi < b \end{array}$$

19.10.3 Romberg integration

Let us compute the integration of $f(x)$ using the trapezoidal rule over the interval $[a, b]$ with $\Delta x = h$. If we denote the output of the trapezoidal rule as $T_{0,1}$, then

$$\text{True value} = T_{0,1} + O(h^2) \tag{19.96}$$

Now let us assume that $O(h^2) = C_1 h^2$, where C_1 is constant. Then,

$$\text{True value} = T_{0,1} + C_1 h^2 \tag{19.97}$$

If we double the number of subintervals such that $\Delta x = \frac{h}{2}$, then,

$$\text{True value} \approx T_{1,1} + C_1 \left(\frac{h}{2}\right)^2 \tag{19.98}$$

There are two unknowns in equation (19.97) and (19.98), *True value* and constant C_1. Subtracting (19.97) from four times (19.98) yields

$$\text{True value} \approx T_{0,2} \equiv T_{1,1} + \frac{1}{3}(T_{1,1} - T_{0,1}) \tag{19.99}$$

Similarly, we can obtain for $\Delta x = \frac{h}{4}$

$$\text{True value} \approx T_{2,1} + \frac{C_1}{16}h^2 \tag{19.100}$$

From (19.98) and (19.100), we obtain

$$T_{1,2} = T_{2,1} + \frac{1}{3}(T_{2,1} - T_{1,1}) \tag{19.101}$$

We can make a further improvement by using $T_{0,2}$ and $T_{1,2}$ and setting up the relations

$$\text{True value} \;=\; T_{0,2} + C_2 h^4 \tag{19.102}$$

$$\text{True value} \;\approx\; T_{1,2} + C_2 \left(\frac{h}{2}\right)^4 \tag{19.103}$$

and hence

$$T_{0,3} = T_{1,2} + \frac{1}{15}(T_{1,2} - T_{0,2}) \tag{19.104}$$

$T_{0,1}$	$T_{0,2}$	$T_{0,3}$
$T_{1,1}$	$T_{1,2}$	
$T_{2,1}$		

We can arrange procedures in a matrix form as
We can also summarize the rule as

$$\text{Improved value} = \text{more accurate} + \left(\frac{1}{2^n - 1}\right)(\text{more accurate} - \text{less accurate})$$

where n is the exponent on h in the error term $O(h^n)$.

Examples: $\int_0^1 \cos(x)dx = \sin(1) = 0.841470985$

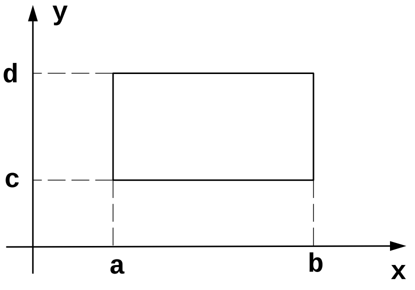
$$\begin{aligned}
T_{0,1} &= \frac{1}{2}(\cos(0) + \cos(1)) = 0.770151153 \\
T_{1,1} &= \frac{1}{2}T_{0,1} + \frac{1}{2}\cos(0.5) = 0.823866857 \\
T_{2,1} &= \frac{1}{2}T_{1,1} + \frac{1}{4}(\cos(0.25) + \cos(0.75)) = 0.837083751 \\
T_{0,2} &= T_{1,1} + \frac{1}{3}(T_{1,1} - T_{0,1}) = 0.841651966 \\
T_{1,2} &= T_{2,1} + \frac{1}{3}(T_{2,1} - T_{1,1}) = 0.841489382 \\
T_{0,3} &= T_{1,2} + \frac{1}{16}(T_{1,2} - T_{0,2}) = 0.841479221
\end{aligned}$$

which is accurate to 5 significant digits.

19.10.4 Double integrals

$$\iint_A f(x,y)dA \tag{19.105}$$

1. *Over a rectangular domain*



$$I = \iint_A f(x,y)dA = \int_a^b \left(\int_c^d f(x,y)dy\right)dx = \int_c^d \left(\int_a^b f(x,y)dx\right)dy \tag{19.106}$$

We will use the trapezoid rule in x and y directions. Let

$$\begin{aligned} h_x &= \frac{b-a}{n} && (19.107) \\ h_y &= \frac{d-c}{n} && (19.108) \end{aligned}$$

we start with $y = c$,

$$\begin{aligned} y = c : I_1 &= \int_a^b f(x, c)dx \\ &= \frac{h_x}{2}(f_1(c) + 2f_2(c) + \cdots + 2f_n(c) + f_{n+1}(c)) && (19.109) \end{aligned}$$

where $f_i(c) = f(x_i, c)$. Similarly,

$$\begin{aligned} y = c + h_y : I_2 &= \int_a^b f(x, c + h_y)dy \\ &= \frac{h_x}{2}(f_1(c + h_y) + 2f_2(c + h_y) + \cdots + 2f_n(c + h_y) + f_{n+1}(c + h_y)) \\ & && (19.110) \\ \cdots \\ y = d : I_{n+1} &= \int_a^b f(x, d)dy \\ &= \frac{h_x}{2}(f_1(d) + 2f_2(d) + \cdots + 2f_n(d) + f_{n+1}(d)) && (19.111) \end{aligned}$$

We now sum $I_1, I_2, \cdots, I_{n+1}$ in y direction in terms of the trapezoidal rule

$$I = \frac{h_y}{2}(I_1 + 2I_2 + \cdots + 2I_n + I_{n+1}) \qquad (19.112)$$

2. *Over a curved boundary domain*, see Figure 19.15

   First we need to find a_i, b_i. We equate $y(t)$ with $c + (i-1)h_y$, where $c = min \;\; y(t)$. Then we solve for t, leading to two (or more) intersections. Plugging the resulting t into $x(t)$ yields a_i and b_i.

   Similar to the rectangular domain case, we use the trapezoidal rule in both x and y directions.

   Let

$$\begin{aligned} h_x^i &= \frac{b_i - a_i}{n} && (19.113) \\ h_y &= \frac{d-c}{n} && (19.114) \end{aligned}$$

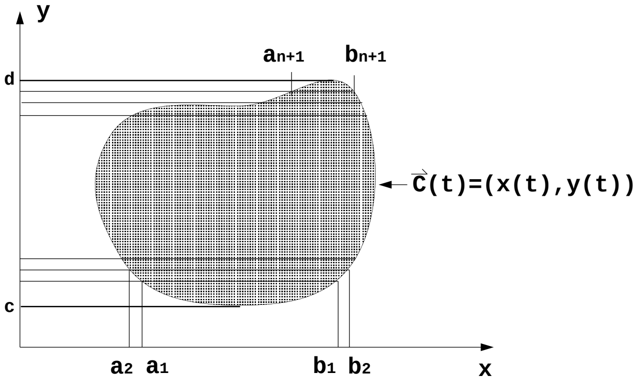


Figure 19.15: A curved boundary domain

We start with $y = c$:

$$\begin{aligned} y = c : I_1 &= \int_{a_1}^{b_1} f(x, c) dx \\ &= \frac{h_x^1}{2}(f_1(c) + 2f_2(c) + \cdots + 2f_n(c) + f_{n+1}(c)) \qquad (19.115) \\ y = c + h_y : I_2 &= \int_{a_2}^{b_2} f(x, c + h_y) dx \\ &= \frac{h_x^2}{2}(f_1(c + h_y) + 2f_2(c + h_y) + \cdots + 2f_n(c + h_y) + f_{n+1}(c + h_y)) \\ & \qquad (19.116) \\ \cdots \\ y = d : I_{n+1} &= \int_{a_{n+1}}^{b_{n+1}} f(x, d) dx \\ &= \frac{h_x^{n+1}}{2}(f_1(d) + 2f_2(d) + \cdots + 2f_n(d) + f_{n+1}(d)) \qquad (19.117) \end{aligned}$$

These formulae can be extended to curved domain boundaries bounding multiply connected domains.

Bibliography

[1] L. Bardis and N. M. Patrikalakis. Topological structures for generalized boundary representations. MITSG 94-22, MIT Sea Grant College Program, Cambridge, Massachusetts, September 1994.

[2] E. Brisson. Representing geometric structures in d dimensions: Topology and order. *Discrete and Computational Geometry*, 9:387–426, 1993.

[3] G. Dahlquist and Å. Björck. *Numerical Methods.* Prentice-Hall, Inc., Englewood Cliffs, NJ, 1974.

[4] D. P. Dobkin and M. J. Laszlo. Primitives for the manipulation of three-dimensional subdivisions. In *Proceedings of the Third ACM Symposium on Computational Geometry*, pp. 86–99, Waterloo, Canada, June 1987.

[5] D. P. Dobkin and M. J. Laszlo. Primitives for the manipulation of three-dimensional subdivisions. *Algorithmica*, 4:3–32, 1989.

[6] L. Guibas and J. Stolfi. Primitives for the manipulation of general subdivisions and the computation of Voronoi diagrams. *ACM Transactions on Graphics*, 4(2):74–123, April 1985.

[7] F. B. Hildebrand. *Advanced Calculus for Applications.* Prentice-Hall, Inc., Englewood Cliffs, New Jersey, 1976.

[8] C. M. Hoffmann. *Geometric and Solid Modeling: An Introduction.* Morgan Kaufmann Publishers, Inc., San Mateo, California, 1989.

[9] G. M. Hunter and K. Steiglitz. Operations on images using quad trees. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1(2):145–153, 1979.

[10] Y. T. Lee and A. A. G. Requicha. Algorithms for computing the volume and other integral properties of solid objects, I: Known methods and open issues. *Communications of the ACM*, 25(9):635–641, September 1982.

[11] M. Mäntylä. *An Introduction to Solid Modeling.* Computer Science Press, Rockville, Maryland, 1988.

[12] D. Meagher. Geometric modeling using octtree encoding. *Computer Graphics and Image Processing*, 19:129–147, June 1982.

[13] M. E. Mortenson. *Geometric Modeling.* John Wiley and Sons, New York, 1985.

[14] A. A. G. Requicha. Representations of solid objects - theory, methods and systems. *ACM Computing Surveys*, 12(4):437–464, December 1980.

[15] V. Shapiro. Solid modeling. In G. Farin *et al.*, editor, *Handbook of Computer Aided Geometric Design, Chapter 20*, pp. 473–518. Elsevier, Amsterdam, 2002.

[16] R. Yagel, D. Cohen, and A. Kaufman. Context sensitive normal estimation for volume imaging. In N. M. Patrikalakis, editor, *Scientific Visualization of Physical Phenomena*, pp. 211–232. Tokyo: Springer-Verlag, 1991.