ADAPTIVE TRIANGULATIONS

by

Oleksandr Maizlish

A Thesis submitted to the Faculty of Graduate Studies of
The University of Manitoba
in partial fulfilment of the requirements of the degree of

DOCTOR OF PHILOSOPHY

Department of Mathematics
University of Manitoba
Winnipeg

UNIVERSITY OF MANITOBA
DEPARTMENT OF MATHEMATICS

The undersigned hereby certify that they have read and recommend to the Faculty of Graduate Studies for acceptance a thesis entitled "***Adaptive triangulations***" by *Oleksandr Maizlish* in partial fulfillment of the requirements for the degree of *Doctor of Philosophy*.

Dated: _____

Research Supervisor: _____
Kirill Kopotun

Internal Examiner: _____
Nina Zorboska

Internal Examiner: _____
Gabriel Thomas

External Examiner: _____
Carl de Boor

UNIVERSITY OF MANITOBA

Date: *June 2013*

Author: *Oleksandr Maizlish*

Title: ***Adaptive triangulations***

Department: *Mathematics*

Degree: *Ph.D.*

Convocation: *May*

Year: *2014*

Permission is herewith granted to University of Manitoba to circulate and to have copied for non-commercial purposes, at its discretion, the above title upon the request of individuals or institutions.

Signature of Author

To my brother Leonid,

who has always been an example for me

Abstract

In this dissertation, we consider the problem of piecewise polynomial approximation of functions over sets of triangulations. Recently developed adaptive methods, where the hierarchy of triangulations is not fixed in advance and depends on the local properties of the function, have received considerable attention. The quick development of these adaptive methods has been due to the discovery of the wavelet transform in the 1960's, probably the best tool for image coding.

Since the mid 80's, there have been many attempts to design 'Second Generation' adaptive techniques that particularly take into account the geometry of edge singularities of an image. But it turned out that almost none of the proposed 'Second Generation' approaches are competitive with wavelet coding. Nevertheless, there are instances that show deficiencies in the wavelet algorithms. The method suggested in this dissertation incorporates the geometric properties of convex sets in the construction of adaptive triangulations of an image. The proposed algorithm provides a nearly optimal order of approximation for cartoon images of convex sets, and is based on the idea that the location of the centroid of certain types of domains provides a sufficient amount of information to construct a 'good' approximation of the boundaries of those domains. The new algorithm is presented in Chapter 2 with the main result established in Theorem 2.2. Along with the theoretical analysis of the algorithm, a Matlab code has been developed and implemented on some simple cartoon images.

Acknowledgments

I wish to thank, first and foremost, my advisor Prof. Kirill Kopotun for sharing his knowledge with me; I thank him as well for his patience and encouragement throughout my graduate studies.

I express my gratitude to Prof. Carl de Boor for his many valuable comments that improved the presentation of this dissertation.

I would also like to acknowledge the Natural Sciences and Engineering Research Council of Canada for its support during the years of my Ph.D. study.

And of course, this work would not be possible without the support and inspiration from my Ukrainian and Canadian families.

Contents

List of Figures

0.1 Notations

In this section, we list the notations used throughout the dissertation. The notations introduced in the subsequent chapters are listed with a reference to the page they are defined on.

Set Theory:

Given a subset S of a metric space (X, ρ),

- $\overline{S}$ denotes the *closure* of S;
- $\text{int}(S)$ denotes the *interior* of S;
- ∂S denotes the *boundary* of S, $\partial S = \overline{S} \backslash \text{int}(S)$.
- $\text{dist}(x, S) := \inf_{y \in S} \rho(x, y)$ denotes the *distance* between a point $x \in X$ and the set S.

Planar Geometry:

- AB denotes the *straight line segment* between the points A and B in the plane.
- $|AB|$ denotes the *length* of the straight line segment AB, and, in general, $|\gamma|$ denotes the *arc length* of a rectifiable curve γ.
- $\overrightarrow{(A, B)}$ denotes the *ray* (half-infinite line) originating at A and passing through point B.
- $\overrightarrow{AB}$ denotes the *vector* on the plane connecting the initial point A with the terminal point B.
- $\overleftrightarrow{AB}$ denotes the *line* on the plane passing through the points A and B.
- $\angle A$ denotes the *angle*, *i.e.*, the geometric figure formed by two rays (called the *sides* of the angle), originating at the common *vertex* A.
- $S(\angle A)$ denotes the set of points inside $\angle A$ together with the sides of the angle (see page 38).

- $\Gamma(\angle A)$ denotes the set of *simple rectifiable* curves inside $S(\angle A)$ with the end-points on the sides of $\angle A$ (see page 38).

- $\Gamma_0(\angle A)$ denotes the set of *convex* curves $\gamma \in \Gamma(\angle A)$ that have two points of intersection with its straight line segment approximation $DE = DE(\angle A, \gamma)$ (see pages 46,50).

- $\Phi_{\angle A}(\gamma)$ denotes the *'angular set'* for each $\gamma \in \Gamma(\angle A)$ (see page 38).

- Δ (or $\triangle$) denotes a *triangle*, *i.e.*, a closed region bounded by three straight line segments (called the *sides* of the triangle) connecting three points on the plane (called the *vertices* of the triangle). A triangle with vertices A, B and C is denoted by $\triangle ABC$. The interior angles $\angle A, \angle B$ and $\angle C$ of $\triangle ABC$ can be also denoted by $\angle BAC, \angle ABC$ and $\angle ACB$, respectively.

- $\mathrm{conv}(S)$ denotes the *convex hull* of a set $S \subset \mathbb{R}^2$ (see page 29).

Function Theory:

Given an open subset U of the Euclidean space $\mathbb{R}^m$, $m \geq 1$,

- $\mathbb{P}_n$ denotes the set of *polynomials* of degree $\leq n$.

- $C(U)$ denotes the set of all *continuous* real-valued functions on U.

- $C^{(k)}(U)$ denotes the set of all k *times continuously differentiable* real-valued functions f on U, *i.e.*, all of the (partial) derivatives of order k of f exist and are continuous.

- $d^2 f$ denotes the *Hessian* matrix of f, *i.e.*, for $f \in C^{(2)}(U)$, $d^2 f$ is a $m \times m$ matrix such that $(d^2 f)_{ij} = \dfrac{\partial^2 f}{\partial x_i \partial x_j}$, $1 \leq i, j \leq m$.

- $\mathrm{supp}(f)$ denotes the *support* of the function f, *i.e.*, the smallest closed set such that f is identically zero almost everywhere outside this set.

- $\lambda = \lambda_m$ denotes the *Lebesgue* measure in $\mathbb{R}^m$.

- $L_{1,loc}(U)$ denotes the set of *locally integrable* real-valued functions on U, *i.e.*, (Lebesgue) integrable on any compact subset $K \subset U$.

- $L_p(U)$, $0 < p \leq \infty$, denotes the space of all measurable functions $f : U \to \mathbb{R}$ such that

$$\|f\|_{L_p(U)} := \left\{ \begin{array}{ll} \left(\int_U |f(x)|^p d\lambda_m(x) \right)^{1/p}, & \text{if } p < \infty, \\ \operatorname{ess\,sup}_{x \in U} |f(x)|, & \text{if } p = \infty \end{array} \right\} < \infty.$$

- l_p, $0 < p \leq \infty$, denotes the space of all sequences $\{x_n\}_{n=1}^{\infty}$ such that

$$\|x\|_p := \left\{ \begin{array}{ll} \left(\sum_{n=1}^{\infty} |x_n|^p \right)^{1/p}, & \text{if } p < \infty, \\ \sup_{n \geq 1} |x_n|, & \text{if } p = \infty \end{array} \right\} < \infty.$$

- The function $g \in L_{1,loc}(U)$ is called the α^{th}*-weak derivative* of $f \in L_{1,loc}(U)$ (α is a multi-index) if

$$\int_U f D^\alpha \varphi = (-1)^{|\alpha|} \int_U g\varphi$$

holds for all infinitely many times differentiable functions φ with compact support in U. Notation: $g = D^\alpha f$.

- $W^{k,p}(U)$ denotes the *Sobolev space*, *i.e.*, the set of all functions $f \in L_p(U)$ such that for every multi-index α with $|\alpha| \leq k$, $D^\alpha f \in L_p(U)$.

- Let $\Delta_h^r(f, x)$ denote the r*th order difference* with step $h \in \mathbb{R}^m$, *i.e.*,

$$\Delta_h^r(f, x) = \sum_{k=0}^{r} \binom{r}{k} (-1)^{r-k} f(x + kh).$$

Then the *modulus of smoothness* of order r of $f \in L_p(U)$, $p > 0$ is defined as

$$\omega_r(f, t)_p := \sup_{|h| \leq t} \|\Delta_h^r(f, \cdot)\|_{L_p(U(rh))}, \quad t > 0,$$

where $\|\cdot\|_{L_p(U(rh))}$ is a (quasi-)norm on $L_p(U(rh))$, with $U(rh) := \{x \in U : [x, x + rh] \subset U\}$.

- $BV([0,1]^2)$ denotes the space of the functions of *bounded variation* on the unit square $[0,1]^2$ (see the details on page 16).

- $B^s_{p,q}$ $(s,p,q>0)$ denotes the *Besov space*, *i.e.*, the set of functions $f \in L_p(U)$ such that for $s<r$,

$$|f|_{B^s_{p,q}} = \left(\int_0^\infty (t^{-s}\omega_r(f,t)_p)^q \frac{dt}{t} \right)^{1/q}$$

  is finite.

- χ_Y is the characteristic function of a set $Y \subset \mathbb{R}^m$, *i.e.*,

$$\chi_Y(x) = \begin{cases} 1, \text{ if } x \in Y, \\ 0, \text{ otherwise.} \end{cases}$$

- A function that is smooth on a finite number of subdomains $Y_i \subset \mathbb{R}^2$ separated by a union of smooth discontinuity curves is called a *cartoon* function (image).

Discrete Mathematics and Graph Theory:

- $\#S$ denotes the *cardinality* of a set S.

- $G=(V,E)$ denotes an *undirected graph*, *i.e.*, an ordered pair (V,E), where the first component is a non-empty set V of the *vertices* of the graph, and the second component is a set E of 2-element subsets of V (the so-called *edges*).

- Given a graph $G=(V,E)$, a *path* in graph G is defined as a sequence of edges that connect a sequence of vertices. A path is called *simple* if it has no repeated vertices. The *length* of a path is the number of edges that the path uses (counting multiple edges multiple times).

- A *tree* is an undirected graph such that any two of its vertices can be connected by exactly one simple path. A tree is called *rooted* if one vertex has been designated the root, in which case the edges have a natural orientation, towards or away from the root.

- A *forest* is a disjoint union of trees.
- Let $\{a_n\}_{n=1}^{\infty}$ and $\{b_n\}_{n=1}^{\infty}$ be two sequences of real numbers. Then we say that

$$a_n = O(b_n), \quad n \to \infty,$$

if there exist a natural number n_0 and a positive constant C such that

$$|a_n| \leq C|b_n|, \quad \text{for all } n \geq n_0.$$

Let $a_n, b_n \geq 0$. We say that $a_n \sim b_n$ as $n \to \infty$ (a_n and b_n are asymptotically equal) if

$$a_n = O(b_n),\ b_n = O(a_n), \qquad n \to \infty.$$

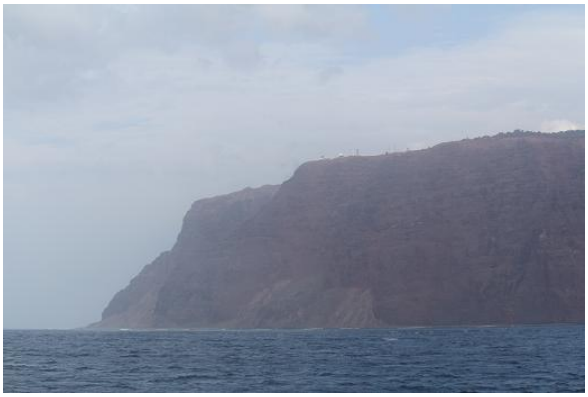
Chapter 1

Geometric Methods in Image Coding

1.1 Introduction

Until the 1980's, most image coding methods relied on techniques based on classical information theory and exploited the redundancy in the images in order to achieve compression. These techniques utilized the information carried by each individual pixel and did not use any specific features of the whole image (see overview in [15],[28],[35]). However, any non-artificial image is a combination of geometric, smooth, and textured regions. Geometric characteristics, so-called *edges*, usually indicate the transitions between smooth or textured regions and are often represented by rapid variance in the pixel intensity (or simply speaking, pixel color) in the neighbourhood of straight or curved contours. Edges communicate important information, conveying the location and shape of pictured objects. Figure 1.1a shows an example of an image where the relevant information is mainly carried along a set of edges (the line of the horizon between the sky and the ocean along with the shape of the mountain). Natural images (see Figure 1.1b) are usually more complex than in the example shown in Figure 1.1a (or any cartoon image), but taking advantage of geometric structures in them is crucial for an efficient compression/processing algorithm.

Figure 1.1: Geometric vs Natural images



(a) Geometrical image

(b) Natural image

Geometric structures appear in various signal models and often carry most of the perceptual information [19]. The motion of objects in a movie is described using an optical flow that follows the $3D$ geometry of the signal ($2D$ picture flow over time). Natural sounds also exhibit geometric patterns in the time-frequency plane where evolving harmonics follow geometric paths. All these geometric features are essential for human perception and should be exploited by modern signal processing methods. While first-generation lossy image techniques could provide high ratios of compression (greater than 30 to 1) only at the expense of image quality [29], second-generation image techniques attempt to identify geometric features of the image and thereby separate the visually significant and insignificant areas of the image and then apply appropriate coding techniques to each area afterwards [18],[28].

From a mathematical perspective, the tools of classical differential geometry can efficiently characterize contours when the edge curves are well defined. Introduction of wavelet transform and wavelet bases allows efficient representation of the regular parts of images. This is the reason why orthogonal wavelet bases are at the heart of JPEG2000 [32], the latest and probably the best image compression standard, which totally supersedes the well-known image standard JPEG. However, in the case of edges where a singularity extends along a contour, the number of $2D$ wavelets overlapping the singularity grows exponentially at higher scales, and thus, reconstruction of even a simple, straight edge requires 'many' wavelet coefficients [34]. The problem

gets even more complex since natural images have varying blurring and turbulent textures, and even the local description of geometric regularity is not very well defined. In Section 1.2, we discuss the wavelet approach and explain its deficiencies for geometric images. In the next sections, we will continue the overview of the geometric techniques in Image Processing in a sequence similar to the structure of [22]; the reader can find information on the modern state of the art in Image Compression techniques, however, we do not aim to mention all of the existing tools and focus only on the basic innovative ideas in Data Compression.

A thresholding in a wavelet basis is equivalent to a finite element approximation with a square support, such that the elements are refined near the singularities. In Section 1.4, we will introduce another square-based representation using wedgelets. In order to geometrically adapt to the edge singularities, the wedgelets of Donoho divide the support of the image in dyadic adapted squares, and on each square, the image is approximated with a constant value on each side of some straight edge. The choice of this edge is optimized using the local content of the image. This approach is generalized by Shukla et al. [31] where they replace constant values by polynomials of higher degree and the straight edges by polynomial curves. This approach is efficient as long as the geometry of the image is again not too complex and edges are not blurred.

To enhance the performance of the finite element method, it is necessary to use geometrically more flexible figures; for instance, an adaptive triangulation. In Section 1.3, we give an overview of another representation method proposed by Candes and Donoho that uses so-called curvelets, functions with support elongated along the singularities (similar to thin and long triangles stretched along the contours of an image).

In Section 1.5, an overview of general adaptive isotropic and anisotropic triangulation methods is given (with no fixed hierarchy of triangles used in representations). We discuss the optimality of the Newest Vertex Bisection Method among adaptive isotropic triangulation methods as well as progress in the anisotropic direction.

Finally, in Chapter 2, we suggest a completely new and geometric approach. The

main idea of the method is that the location of the centroid of an area bounded by two sides of a triangle and a curve crossing these two sides contains a sufficient amount of information to construct a 'good' approximation to the curve. Furthermore, for a special class of curves, knowing the coordinates of the centroid allows full reconstruction of the curve. The condition of convexity is a significant part of the algorithm thus far, as there are many nice and simple facts regarding the centroid of a convex body along with more complicated but no less beautiful results like Winternitz's theorem [3, pp. 54-55]. We show that the proposed algorithm implies a nearly optimal order of approximation for cartoon images.

1.2 Wavelet Coding

A general transform coding scheme involves subdividing an $N \times N$ image into smaller $n \times n$ blocks and performing a linear invertible transform on each sub-image. This discrete transformation produces a representation in a new, often orthonormal basis. This procedure generally results in the signal energy being redistributed among only a small set of the coefficients in the new basis representation. For instance, in the JPEG compression standard, each 8×8 block is converted to a frequency-domain representation using the 2-dimensional Discrete Cosine Transform. As a result, most of the signal has a tendency to aggregate in one corner of an 8×8 frequency-domain representation of the sub-image. This step is followed by *quantization*, when the signal is being multiplied by some scale and then rounded to the nearest integer. The procedure of quantization in JPEG is the only lossy operation in the process: it produces many zeroes in the higher frequency coefficients and thus reduces the number of bits required to represent the signal.

Now given a continuous (periodic) signal $f(t)$, decomposing it in an orthogonal basis allows one to define a sparse representation using a simple thresholding. In particular, one can decompose the signal into the superposition of its high order harmonics based on the Fourier expansion theory. The following example shows how a truncated Fourier series approximates the original signal.

Example 1.1. Let f be a T-periodic function ($T > 0$) such that $f(t) = 1 - 2|t|/T$, for $|t| < T/2$ (see Figure 1.2).

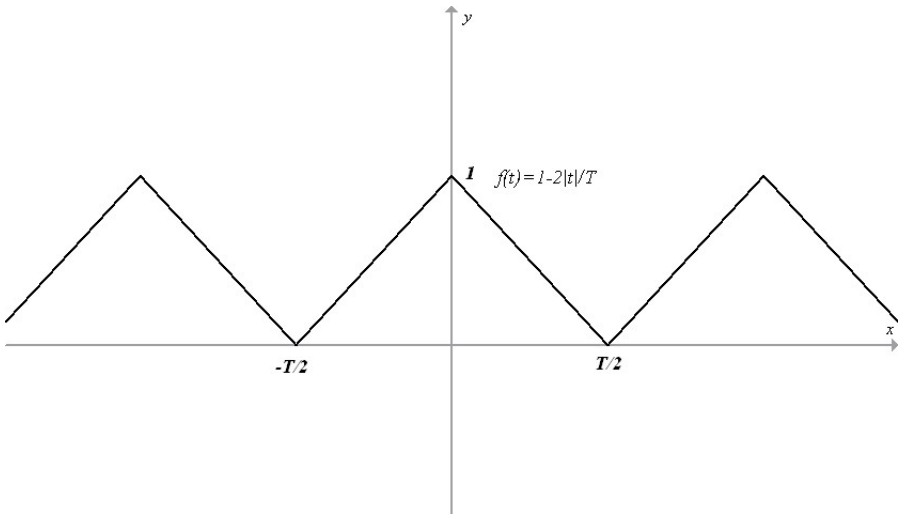


Figure 1.2: Periodic signal

Then, the Fourier series representation for f is

$$f(t) = \sum_{k=-\infty}^{\infty} a_k \exp\left(i\frac{2\pi kt}{T}\right) = a_0 + \sum_{k=1}^{\infty} 2a_k \cos\left(\frac{2\pi kt}{T}\right),$$

where the Fourier coefficients a_k are

$$a_k = \frac{1}{T}\int_{-T/2}^{T/2} f(t)\exp\left(-i\frac{2\pi kt}{T}\right)\,dt = \frac{2\sin^2(\pi k/2)}{(\pi k)^2},$$

and we use the fact that f is even.

Based on the decreasing rate of the coefficients, the signal concentrates most in the low frequency components. Below on Figure 1.3, we show the approximation of the signal f obtained by truncating the series coefficients in the range $|k| \leq 1$, *i.e.*, by keeping only the terms in the series that correspond to $k = -1, 0, 1$.
In general, as N approaches infinity, we can expect the approximation

$$\sum_{|k|\leq N} a_k \exp\left(i\frac{2\pi kt}{T}\right)$$

to be arbitrarily close to the original signal.

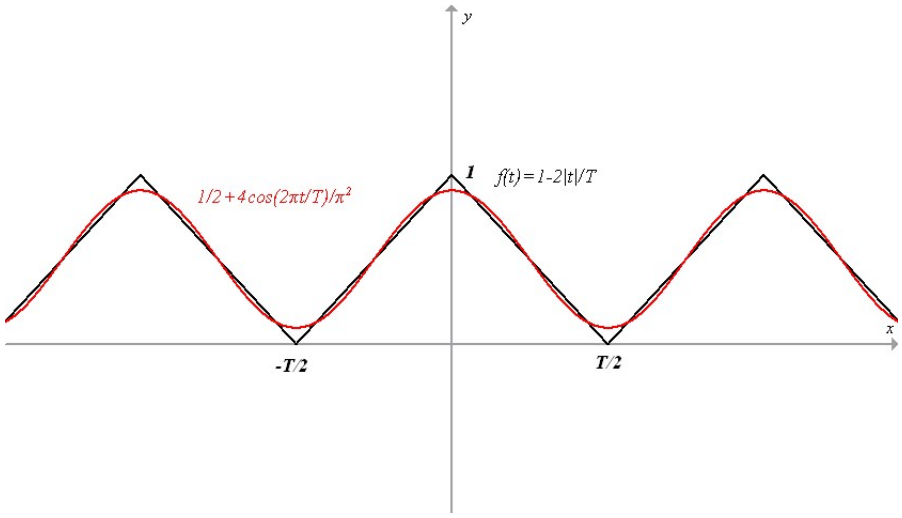


Figure 1.3: Approximating signal using truncated Fourier series with $|k| \leq 1$

However, if we take a Fourier transform of not necessarily periodic functions over the infinite time domain, we cannot distinguish at what instant a particular frequency occurs. Time-localization can be partially resolved by first windowing the signal [8], and then taking its Fourier transform. A full solution to this problem can be achieved by introducing a different basis for representation, a so-called wavelet basis that can localize both the time and frequency of the signal.

The first reference to the idea of wavelet bases was proposed by the mathematician Alfred Haar [14] in 1910. However, the concept of the wavelet did not exist at that time. In 1981, the concept was proposed by geophysicist Jean Morlet. Afterwards, Morlet and physicist Alex Grossman invented the term wavelet in 1984 (or to be precise, they used an equivalent French word 'ondelette', meaning 'small wave'). Since then, a wavelet system usually involves dilations and shifts of a single function that form an orthogonal basis in some space. Before 1985, the Haar wavelet was the only existing orthogonal wavelet until the mathematician Yves Meyer constructed a second orthogonal wavelet system now called Meyer wavelets [23] in 1985. More and more scholars joined this study at the first international conference held in France in 1987. In 1988, Stephane Mallat and Meyer proposed the concept of multiresolution analysis. In the same year, Ingrid Daubechies found a systematical method to construct compactly supported orthogonal wavelets. In 1989, Mallat proposed

the fast wavelet transform. With the appearance of this fast algorithm, the wavelet transform received numerous applications in Signal Processing.

Returning to the basis representation problem, given $f \in L_2(I)$ (I is a possibly infinite interval), the best approximation f_N of the function f with N coefficients in an orthonormal basis $\mathcal{B} = \{g_\mu\}_{\mu \in S}$ is computed using N absolutely largest coefficients above some threshold L:

$$f_N = \sum_{|\langle f, g_\mu \rangle| > L} \langle f, g_\mu \rangle g_\mu, \quad N := \#\{\mu : |\langle f, g_\mu \rangle| > L\},$$

where $\langle \cdot, \cdot \rangle$ denotes the inner product in $L_2(I)$. It is easy to see that then the approximation error is

$$\|f - f_N\|^2_{L_2(I)} = \sum_{|\langle f, g_\mu \rangle| \leq L} |\langle f, g_\mu \rangle|^2.$$

Hence, aiming to optimize the representation, one is looking for a basis $\mathcal{B}$ such that the decay of the approximation error is maximized, *i.e.*, $\|f - f_N\|^2_{L_2(I)} = O(N^{-\beta})$, $N \to \infty$, for the largest possible β. Thus, the approximation problem is the key part of any compression algorithm.

Remark 1.1. *In some instances, we may consider expansions in more general systems of functions:*

- **Frames.** *Given a vector space V with inner product $\langle \cdot, \cdot \rangle$, a sequence of vectors $\{v_k\}$ is called a frame if there exist real A and B, $0 < A \leq B < \infty$, such that*

$$A\|u\|^2 \leq \sum_k |\langle u, v_k \rangle|^2 \leq B\|u\|^2, \quad \text{for all } u \in V. \tag{1.1}$$

One can show that condition (1.1) *implies existence of a sequence of dual frame vectors $\{\tilde{v_k}\}$ such that, for any $u \in V$,*

$$u = \sum_k \langle u, \tilde{v_k} \rangle v_k = \sum_k \langle u, v_k \rangle \tilde{v_k}.$$

A frame is called tight if $A = B$ in condition (1.1)*, i.e., Parseval's identity is satisfied.*

A frame is called exact if it ceases to be a frame whenever any single element is removed from it.

- **Riesz Basis.** *Given a Hilbert space H, a system $\{x_k\} \subset H$ is a Riesz system with constants $A, B > 0$ if for any $c = \{c_k\} \in l_2$, the series $\sum_k c_k x_k$ converges in H and*

$$A\|c\|_{l_2}^2 \leq \left\| \sum_k c_k x_k \right\|_H^2 \leq B\|c\|_{l_2}^2.$$

If additionally the system $\{x_k\}$ is a basis, it is called a Riesz basis. Note that any sequence $\{x_k\}$ in a Hilbert space H is an exact frame for H if and only if it is a Riesz basis in H.

We refer readers to, for example, [27, §1.1,1.8] *for more information about frames and Riesz bases.*

1D-wavelets. Given a function $\psi \in L_2(\mathbb{R})$, a system of functions

$$\{\psi_{j,k} = 2^{j/2}\psi(2^j x - k) : j, k \in \mathbb{Z}\}$$

is a called a discrete **wavelet** system if it is an orthonormal basis of $L_2(\mathbb{R})$. In order to characterize approximation smoothness classes, one often requires ψ (also called *mother*-wavelet) to have compact support and r vanishing moments, *i.e.,*

$$\int_{\mathbb{R}} x^l \psi(x)\, dx = 0, \quad 0 \leq l \leq r.$$

This condition arises naturally since a locally smooth function is locally 'well-approximated' by polynomials. And hence, provided ψ has r vanishing moments, for any $f \in L_2(\mathbb{R})$

$$\int_{\mathbb{R}} f\psi_{j,k}\, dx = \int_{\mathbb{R}} (f - P)\psi_{j,k}\, dx,$$

as long as P has degree $\leq r$. If P approximates f well on the interval containing the support of $\psi_{j,k}$, then the corresponding wavelet coefficient $\langle f, \psi_{j,k}\rangle$ is small. Daubechies (see [8, §6.3, Theorem 6.3.6] for instance) showed the possibility of constructing a mother-wavelet with compact support, vanishing moments and generating an orthonormal basis of $L_2(\mathbb{R})$. These properties of a mother-wavelet ensure that (reader can be referred to [13, Theorem 37, p.46]) for a function $f \in C^{(n)}(I)$ and a wavelet $\psi_{j,k}$ with $\text{supp}(\psi_{j,k}) \subset I$,

$$\langle f, \psi_{j,k}\rangle = O(2^{-j(n+1/2)}), \quad j \to \infty \quad \text{(uniformly in } k\text{)}. \tag{1.2}$$

As a corollary from this, one can show that

Corollary 1.1. *If f is a piecewise $C^{(n)}$-function with a finite number of singularities, then $\|f - f_N\|^2_{L_2(\mathbb{R})} = O(N^{-2n})$, $N \to \infty$.*

This asymptotic decay is optimal and coincides with the rate of approximation for an f with no singularity. Hence, the existence of a finite number of singularities does not affect the asymptotic precision of a wavelet approximation on the real line [22].

Figure 1.4 shows a piecewise regular function together with its Haar wavelet coefficients $\langle f, \psi_{j,k} \rangle$, $j = 0, \ldots, 9$. We can notice that the large coefficients are localized in the neighborhood of singularities. At the end, we plot f_N computed with the 10% largest wavelet coefficients. As we can now see, the existence of singularities does not affect the asymptotic precision of a wavelet approximation in this case.

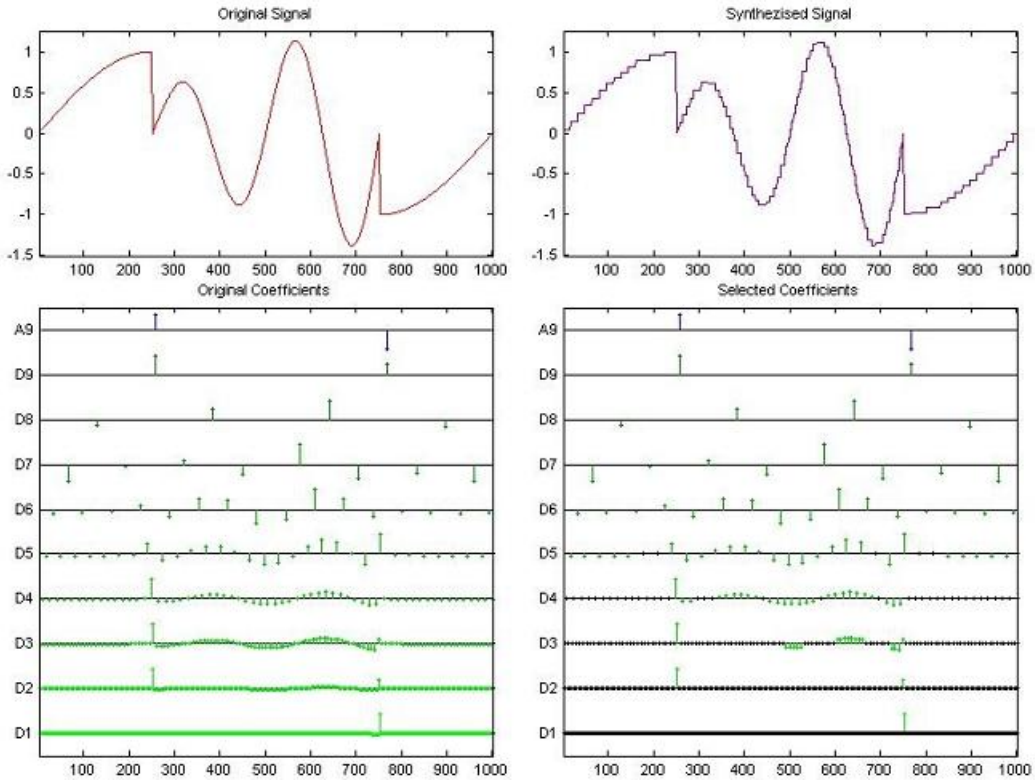


Figure 1.4: Piecewise regular function together with its wavelet coefficients; truncated Haar wavelet representation using 10% largest wavelet coefficients.

2D-wavelets. Wavelets in $\mathbb{R}^2$ can be constructed using tensor products, or more

precisely, by dilations and translations of three elementary wavelets

$$\{\psi^H(x,y), \psi^V(x,y), \psi^D(x,y)\},$$

which oscillate in the horizontal, vertical and diagonal directions (for details, see [8],[27]). These functions may be obtained as a product of the corresponding wavelet functions in the 1-dimensional case of x and y variables, *i.e.*, these wavelets are separable products of monodimensional wavelet functions. Thus,

$$\mathcal{B} = \{\psi^q_{j,k,l}(x,y) = 2^j\psi^q(2^jx-k, 2^jy-l),\ q = H,V,D\}_{j,k,l\in\mathbb{Z}}.$$

2-dimensional wavelets play a key role in the JPEG2000 image compression standard decomposing an image in a wavelet basis followed by quantization and encoding that uses the redundancies in the statistical distribution of the image coefficients and optimizes the binary code [32]. Figure 1.5 shows a famous test-image compressed using wavelet decomposition with a threshold of 10% and 2% largest coefficients. As we can see, the compressed image with 10% threshold is fairly accurate.

Another application of wavelet bases on square domains is the denoising of images. The next example shows the same test-image corrupted with a gaussian white noise W of variance $\sigma = 0.03$. One of the possible approaches is to use a convolution with an optimized filter. However, such a method often suppresses a part of the noise but also smooths the image singularities which creates a blurry image. On the other hand, it is shown in Figure 1.6 that by thresholding 1% of the largest wavelet coefficients, and then performing the inverse wavelet transform on the thresholded coefficients, the noise level in the image is significantly reduced in homogeneous regions and edges are better reconstructed because their wavelet coefficients are retained by the thresholding.

In terms of the accuracy of the wavelet approximation in the 2-dimensional case, the regular $C^{(n)}$ images on a domain Ω allow the rate of convergence similar to the 1-dimensional case:

$$\|f - f_N\|^2_{L_2(\Omega)} = O(N^{-n}), \quad N \to \infty.$$

However, in contrast to the 1-dimensional scenario, this estimate is no longer valid if f is discontinuous along some edge. If f is only a piecewise $C^{(n)}$ image (on the sets

Figure 1.5: 'Lena' test-image and its wavelet decomposition with a threshold of 10% and 2% largest coefficients

outside a finite number of contours of finite length), then the rate of convergence becomes

$$\|f - f_N\|^2_{L_2(\Omega)} = O(N^{-1}), \quad N \to \infty. \tag{1.3}$$

This result is a particular case of a more general statement, a *direct Jackson-type estimate*, from [6]. If f belongs to the class $BV([0,1]^2)$ (of bounded variation), then

$$\|f - f_N\|^2_{L_2([0,1]^2)} \leq CN^{-1}|f|_{BV([0,1]^2)},$$

where the constant C is independent of N and f. The space of the functions of bounded variation with the seminorm $|\cdot|_{BV([0,1]^2)}$ is defined as follows: $f \in BV([0,1]^2)$ if and only if

$$|f|_{BV([0,1]^2)} := \sup_{h>0} h^{-1} \left[\|\Delta_{he_1}(f,\cdot)\|_{L_1(Q(he_1))} + \|\Delta_{he_2}(f,\cdot)\|_{L_1(Q(he_2))}\right] < \infty,$$

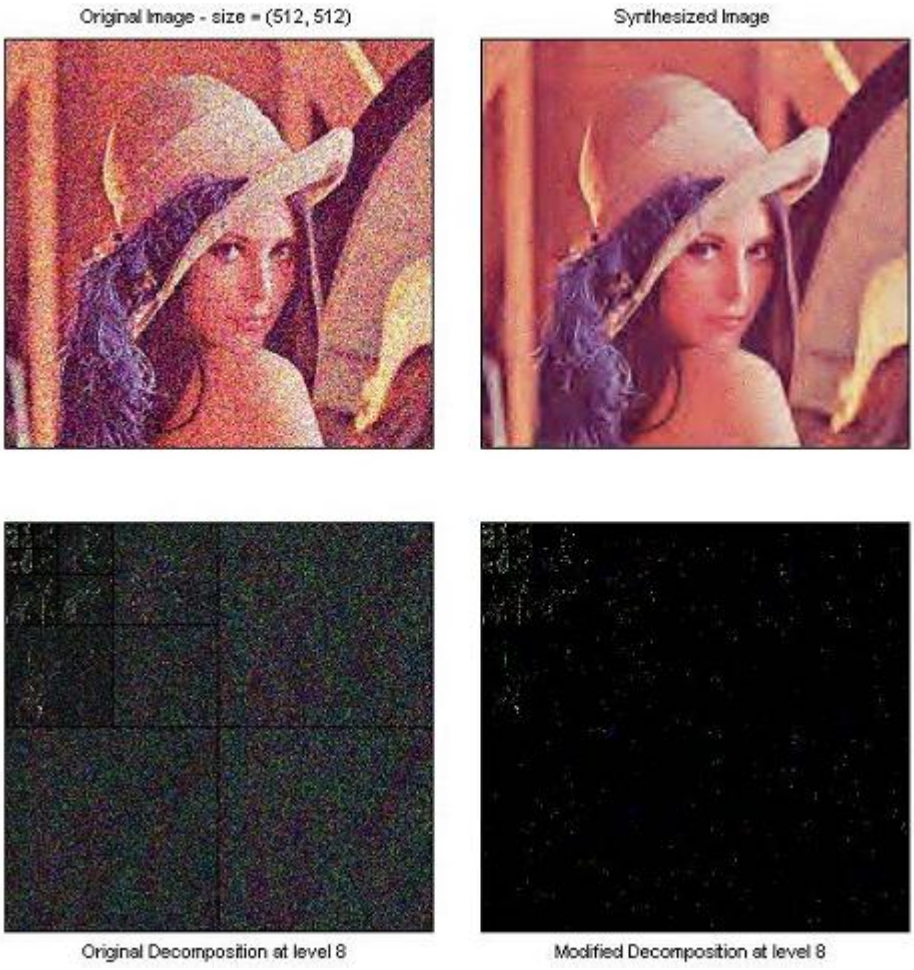


Figure 1.6: 'Lena' test-image corrupted with a gaussian white noise W of variance $\sigma = 0.03$; wavelet denoising using 1% of the largest wavelet coefficients

where $\Delta_\mu(f,x) := \Delta_\mu^1(f,x)$ is the first difference, e_1, e_2 are two unit vectors in x- and y-directions, respectively, and set $Q(\mu) := \{x : [x, x+\mu] \subset [0,1]^2\}$.

However, if the image singularities are located along a smooth contour, one can still hope for an an efficient geometric representation. More precisely, the desired result is that the rate of approximation $\|f - f_N\|^2_{L_2(I)}$ will have the order of N^{-n} as if there were no singularities in the image.

As mentioned in the introduction, thresholding wavelet decomposition by choosing only the N absolutely largest wavelet coefficients is equivalent to using piecewise constant approximation over adaptive anisotropic triangulations with N triangles (or

asymptotically proportional to N). Clearly, we can construct a triangulation with N vertices along the discontinuity edges, and in case of black and white cartoon images with $C^{(2)}$ edges, the approximation f_N^{tr} based on this triangulation will have an error

$$\|f - f_N^{tr}\|_{L_2(\Omega)}^2 \leq CN \cdot N^{-3} = CN^{-2}, \tag{1.4}$$

since the error over each triangle will have an order of N^{-3} due to the estimates of type (2.40) and (2.42) mentioned in Subsection 2.3.3. Thus, estimate (1.4) implies that it is possible to obtain approximation error rates that decay faster than wavelet approximation by adapting the representation to the geometry of the image. Replacing triangles with geometric structures that have polynomial curves as boundaries will lead to even better rates of approximation. However, even in simple cases of triangulation, finding an algorithm that constructs an adaptive triangulation that satisfies (1.4) is still an unresolved problem in general. Some partial progress on this problem will be discussed in Section 1.5.

1.3 Curvelets

Despite the lack of an algorithm to construct result (1.4), it provides an objective performance benchmark. Its asymptotic convergence rate is actually the correct optimal behavior for approximating general smooth objects having singularities along piecewise $C^{(2)}$ curves. Consider a binary (black and white) image for which the curvature of the boundary curve separating black from white is bounded above.

The approximation-theoretic arguments in [11] and [17] lead to the fact that *retaining the N absolutely largest coefficients in any orthogonal basis representation cannot imply an approximation error rate better than N^{-2} (in L_2-norm).* Even if one considers finite linear combinations of N elements of an arbitrary basis (not necessarily orthogonal basis or near-orthogonal system), there is no depth-search-limited dictionary that can achieve a better rate than N^{-2}; see [11]. By "depth-search-limited", we mean allowing sequences of dictionaries whose size grows polynomially in the number of terms to be kept in the approximation. Furthermore, no fixed basis

even comes close to the optimal convergence rate (1.4). In fact, the wavelet convergence rate is the best published nonadaptive result. However, Candes and Donoho showed that there is a basis whose simple thresholding achieves a nearly optimal rate of convergence, and it is due to the introduction of *curvelets*.

There exist different constructions of curvelets. The earlier introduction of curvelets involves the construction of so-called ridgelets [4]. However, in our further discussion we will refer to the second generation of curvelets and follow [5].

Let μ be a triple (j, l, k), where $j \in \mathbb{N} \cup \{0\}$ is a scale parameter; $l = 0, 1, \ldots, 2^j$ is an orientation parameter; and $k = (k_1, k_2)$, $k_1, k_2 \in \mathbb{Z}$, is a translation parameter. Then the system of curvelets $\{\gamma_\mu\}$ is defined as follows

$$\gamma_\mu(x) = 2^{3j/2}\gamma(D_j R_{\theta_J} x - k_\delta),$$

where

- γ is smooth and oscillatory in the horizontal direction and bellshaped (nonoscillatory) along the vertical direction on $\mathbb{R}^2$ (to give a better idea on the structure of γ, one can think of it as a 2-dimensional wavelet separable in variables x_1 and x_2);
- R_θ is the matrix of the planar rotation by θ radians in the counter-clockwise direction;
- $\theta_J = 2\pi l/2^j l$ is the rotation angle ($J = (j, l)$);
- D_j is a parabolic scaling matrix

$$D_j = \begin{pmatrix} 2^{2j} & 0 \\ 0 & 2^j \end{pmatrix},$$

- k_δ is the translation parameter.

It was shown in [5] that for a certain class of γ, $\{\gamma_\mu\}$ is a tight frame of $L_2([0,1]^2)$, and thus any $f \in L_2([0,1]^2)$ can be reconstructed by $f = \sum_\mu \langle \tilde{\gamma_\mu}, f \rangle \gamma_\mu$. Then, if f

is piecewise $C^{(2)}$ with singularities along $C^{(2)}$ contours, the authors have shown that thresholding of the curvelet expansion leads to nearly optimal results.

To further elaborate, let us first introduce some notations. We say that a set $S \in \mathrm{STAR}^2(A)$ if $S \subset [0,1]^2$ and the boundary of S (or translated S) is bounded by some polar curve $r = \rho(\theta)$, $0 \leq \theta \leq 2\pi$, with $\|\rho\|_\infty \leq \rho_0$ and $\|\rho''\|_\infty \leq A$. As well, $C_0^{(2)}([0,1]^2)$ is defined as the collection of twice continuously differentiable functions supported strictly inside the unit square. We also define the collection of functions $\mathcal{E}(A)$ as follows

$$\mathcal{E}(A) := \{f = f_0 + f_1\chi_S : \ f_0, f_1 \in C_0^{(2)}([0,1]^2),\ S \in \mathrm{STAR}^2(A)\}.$$

We are now ready to state the main result on curvelet approximation optimality.

Theorem 1.1 (Candès, Donoho [5])**.** *Let $a_\mu(f) := \langle \gamma_\mu, f \rangle$ be the curvelet coefficient sequence for a function $f \in \mathcal{E}(A)$ ($A > 0$ is fixed). Let also $|a(f)|(N)$ be the Nth absolutely largest entry in the coefficient sequence $\{a(f)_\mu\}_\mu$. Then*

$$\sup_{f \in \mathcal{E}(A)} |a(f)|(N) \leq CN^{-3/2}(\log N)^{3/2},$$

where the constant $C = C(A, \rho_0)$ depends on A and ρ_0.

Thus, we deduce that under the assumptions of Theorem 1.1 above,

$$\|f - f_N^c\|_{L_2}^2 \leq CN^{-2}(\log N)^3, \tag{1.5}$$

where f_N^c is the N-term approximation of f obtained by extracting from the curvelet series the terms corresponding to the N absolutely largest coefficients. It is also worth noting that the results remain valid if we allow several piecewise $C^{(2)}$ edge curves with finitely many intersections, as demonstrated in [5] as well.

Up to the $\log N$ factor in (1.5), one recovers the result (1.4) obtained using an optimal triangulation, but this time with an algorithmic approach. Unlike an optimal triangulation that has to adapt the aspect ratio of its elements (see Section 1.5), the curvelet basis is *a priori* fixed and the thresholding of the curvelet coefficients is enough to adapt the approximation to the geometry of the image. This

simplicity however has a downside. The curvelet approximation is only optimal for piecewise $C^{(n)}$ functions with $n = 2$, but it is no longer optimal for $n > 2$, or for less regular functions such as functions of bounded variation. Recently (see [30]), it was established that the convergence rates of the curvelet approximation increase when smoothness increases. The main result is that the logarithmic factor in the result above is unnecessary if the function and the edges that separate smooth regions have $C^{(3)}$ smoothness.

Another disadvantage of the proposed method in this section is that as of today, none of the constructed bases of curvelets is an orthogonal basis, which makes them less efficient than wavelets for compressing natural images.

1.4 Wedgelets

Many adaptive geometric representations have been proposed recently with good results in Image Processing. The wedgelets of Donoho [10] segment the support of the image into dyadic adapted squares. On each square, the image is approximated with a constant value on both sides of a straight edge. The direction of this estimated edge is optimized using the local content of the image. This approach is generalized by Shukla et al. [31] by replacing constant values by polynomials and the straight edges by polynomial curves.

Now, let us discuss the construction of wedgelets in details. We will consider so-called "horizon" functions defined on the unit square $[0,1]^2$:

$$f(x) = \chi_{\{(x_1,x_2):x_2 \geq H(x_1),\, x_1 \in [0,1]\}}, \quad \text{for some } H(x).$$

For every level $j \geq 0$, the unit square can be partitioned into the set of dyadic squares:

$$[0,1]^2 = \bigcup_{k_1,k_2 \in \{0,1,\ldots 2^j-1\}} S_j(k_1,k_2) := \bigcup_{k_1,k_2 \in \{0,1,\ldots 2^j-1\}} [k_1 2^{-j}, (k_1+1)2^{-j}] \times [k_2 2^{-j}, (k_2+1)2^{-j}].$$

Fix an integer $n = 2^J$. Then each dyadic square $S_J(k_1,k_2)$, $k_1, k_2 \in 0, 1, \ldots, n-1$, can be viewed as one of n^2 pixels of the image with resolution $n \times n$. In order to

construct some approximation to $H(x)$, we may consider a collection of edge elements (edgels) connecting all possible vertices $(k_1/n, k_2/n)$ of the dyadic partition. This construction will, though, lead to $O(n^4)$ distinct edgels. To avoid the search over this obviously large number of edgels, one can reduce the number by introducing a new subfamily of edgels, so-called *edgelets*. Choose a level of resolution $\delta := 2^{-J-K}$, $K \geq 0$. On the perimeter of each dyadic square $S_j(k_1, k_2, j)$, $0 \leq j \leq J$, mark off a set of equispaced vertices, with distance δ apart, starting at the right upper corner. Denote by $V(S_j)$ the set of these vertices v_{i,S_j} labeling them in a clockwise order. Note that $M_j := \#V(S_j) = 2^{J+K-j+2}$. Now, for given dyadic n and δ, we define the set of edgelets as follows

$$\mathcal{E}_{n,\delta} := \bigcup_{0\leq j\leq J} \bigcup_{S_j} E_\delta(S_j) := \bigcup_{0\leq j\leq J} \bigcup_{S_j} \{e = \overline{v_{i_1,S_j} v_{i_2,S_j}} : 0 \leq i_1, i_2 \leq M_j\}.$$

The set $\mathcal{E}_{n,\delta}$ obviously contains fewer than $O(n^4)$ edgels. Taking into account that $\#E_\delta(S_j) = \binom{M_j}{2}$, we estimate $\#\mathcal{E}_{n,\delta}$:

$$\begin{aligned}\#\mathcal{E}_{n,\delta} &= \sum_{j=0}^{J} \sum_{k_1,k_2\in\{0,1,\ldots,2^j-1\}} \#E_\delta(S_j(k_1,k_2)) = \sum_{j=0}^{J} 2^{2j} \binom{M_j}{2} \\ &\leq \sum_{j=0}^{J} 2^{2j} M_j^2/2 = \sum_{j=0}^{J} 2^{2j} 2^{2J+2K-2j+4}/2 = 8(J+1)\left(2^{-J-K}\right)^{-2} \\ &= 8(\log_2 n + 1)\delta^{-2}.\end{aligned}$$

This in particular implies that if $K = 0$ (*i.e.*, $\delta = 1/n$), then the number of edgels in $\mathcal{E}_{n,\delta}$ has order of $O(n^2 \log n)$.

One of the results concerning the approximation of image contours is as follows:

Theorem 1.2 ([10])**.** *Let H be a continuous "horizon" function, $0 \leq H(t) \leq 1$, $t \in [0,1]$. Let $\Gamma := \{(t, H(t)) : t \in [0,1]\}$ be the associated horizon set in $[0,1]^2$, and suppose that this set can be approximated to within Hausdorff distance ε using at most m edgels with arbitrary vertices. Then this curve may be approximated within Hausdorff distance $\varepsilon + \delta$ using at most $8m \log n$ edgelets from $\mathcal{E}_{n,\delta}$, for $n > 2$, $m \geq 2$.*

The collection of edgelets generates an efficient segmentation of images and is a useful tool to represent the edges in the images. In particular, one can build a basis for the space of "horizon" images. Indeed, for each dyadic square S, its every non-degenerate edgelet $e \in E_\delta(S)$ (not lying entirely on one of the sides of S) divides S into two regions. By $\omega_{e,S}$, we denote the characteristic function of one of these regions, namely, the one containing the segment joining vertices $v_{0,S}$ and $v_{1,S}$. Then let

$$W_\delta(S) := \{\chi_S\} \cup \{\omega_{e,S} : e \in E_\delta(S) \text{ is nondegenerate}\}$$

be the set of functions that bijectively correspond to all the ways of splitting S into two pieces by edgelets (including the special case of not splitting it at all). For a given $n = 2^J$ and dyadic δ, by $\mathcal{W}(n, \delta)$ we denote the set of all wedgelets ω_i such that $\omega_i \in W_\delta(S)$, for some dyadic square S_j, $0 \le j \le J$.

In some sense, the wedgelets constructed above provide near-optimal representations of "horizon" functions.

Definition 1.1. *Let H be a function defined on the interval $[0, 1]$. Then we say*

- $H \in \text{Hölder}^\alpha(C)$, $0 < \alpha \le 1$, *if*

$$|H(x) - H(y)| \le C|x - y|^\alpha, \quad x, y \in [0, 1];$$

- $H \in \text{Hölder}^\alpha(C)$, $1 < \alpha \le 2$, *if*

$$|H'(x) - H'(y)| \le C|x - y|^{\alpha-1}, \quad x, y \in [0, 1].$$

Theorem 1.3 ([10])**.** *Let H be a "horizon" function such that $H \in Horiz^\alpha(C_\alpha, C_1) := \text{Hölder}^\alpha(C_\alpha) \cap \text{Hölder}^1(C_1)$, for some $\alpha \in [1, 2]$, and let $f(x_1, x_2) = \chi_{\{x_2 \ge H(x_1)\}}$ be the corresponding image-function defined on $[0, 1]^2$. In addition, suppose that $n > 2$ and $2 \le m \le n$. Then, there exist $m' \le 8(C_1 + 2)m$ and a collection of m' wedgelets $\omega_i \in \mathcal{W}(n, \delta)$ such that*

$$\|f(x_1, x_2) - \sum_{i=1}^{m'} \omega_i(x_1, x_2)\|^2_{L_2([0,1]^2)} \le C\frac{1}{m^\alpha} + \delta,$$

where the constant C depends only on α.

Taking $m = n$ and $\delta = O(m^{-\alpha})$, one can obtain the following estimate on the approximation error of f by $f_{m'}(x_1, x_2) = \sum_{i=1}^{m'} \omega_i(x_1, x_2)$:

$$\|f - f_{m'}\|^2_{L_2([0,1]^2)} = O(m^{-\alpha}), \quad m \to \infty.$$

In addition, wedgelet descriptions achieve the optimal exponent rate $O(\varepsilon^{-2/\alpha})$ of growth of the number of bits necessary to retain to be sure that the reconstruction of any $f \in \text{Horiz}^{\alpha}(C_1, C_\alpha)$ will have accuracy ε.

The wedgelets are known to be very efficient in noise removal applications. Some extensions of the results are possible (such as considering $H \in B^{\alpha}_{p,q}$ or the cartoon images of star-shaped domains).

Due to the fact that wedgelets are discontinuous functions, the reconstructions they provide generate bad visual artifacts away from the actual boundary being estimated: so-called blocking effects. Such artifacts are not acceptable in the Image Processing context. In addition, since the system of wedgelets is overcomplete there have been various approaches to create efficient computational algorithms for wedgelet coding.

To summarize, wedgelets can be used as long as the geometry of the image is not too complex and the edges are not blurred.

1.5 Adaptive Triangulations

Now we restrict our attention to functions defined on polygonal domains and approximation techniques using finite element methods. To start with, there are two distinct approaches: uniform and adaptive. In the uniform case, all the elements of the mesh have comparable shapes and sizes, while these parameters may vary in the case of an adaptive approximation. Furthermore, the mesh in adaptive methods is not fixed in advance but uniquely constructed for each individual function f to be approximated. The function itself may or may not be fully known (only partially known or even fully unknown as in denoising or in solutions of PDEs). From now on, we will consider only triangular partitions.

Another distinction among adaptive methods is between isotropic and anisotropic triangulations. Each triangle in an isotropic triangulation must satisfy some (regularity) condition that guarantees the triangular element does not differ too much from an equilateral triangle. These conditions can be either (i) a lower bound $\theta_0 > 0$ for every angle of the triangle, or (ii) a lower bound on the aspect ratio

$$\rho_\Delta := \frac{R_\Delta}{r_\Delta} \geq \rho_0 > 0, \tag{1.6}$$

where R_Δ and r_Δ are the radii of the circle with triangle Δ inscribed and the circle inscribed in triangle Δ, respectively. For an anisotropic triangulation, the aspect ratio can be arbitrarily large, or geometrically speaking, long and thin triangles are allowed.

In terms of the approximating elements, in this section we assume that the function f is approximated by a linear function on each triangular element. More precisely, given a triangulation $\mathcal{D}_N$ (comprising N triangles) of a polygonal domain Ω, we approximate f by a piecewise linear function f_N such that $f_N\big|_\Delta$ is a polynomial of degree ≤ 1, for any $\Delta \in \mathcal{D}_N$. We are interested in estimating the asymptotic behaviour of $\|f - f_N\|_{L_p(\Omega)}$, as $N \to \infty$.

Definition 1.2. *The best piecewise-linear approximation error of f over triangulations of cardinality N is defined as*

$$e_N(f)_{L_p(\Omega)} := \inf_{\#\mathcal{D}\leq N} \inf_{f_N|_\Delta \in \mathbb{P}_1,\, \Delta\in\mathcal{D}} \|f - f_N\|_{L_p(\Omega)}.$$

In the next part of this section, we follow the approach presented in [24] to investigate the rate of this approximation in three possible scenarios of triangulations: uniform, isotropic and anisotropic.

Uniform triangulations. From numerical analysis basics, one can always establish the following inequality on the error of approximation by the finite elements method: for $f \in W^{2,p}(\Omega)$ and any triangulation $\mathcal{D}$ with $N := \#\mathcal{D}$ and $h := \max_{\Delta\in\mathcal{D}} \operatorname{diam}(\Delta)$,

$$\|f - f_N\|_{L_p(\Omega)} \leq Ch^2 \|d^2 f\|_{L_p(\Omega)}, \tag{1.7}$$

where C is a positive constant independent of N and f. In the estimate (1.7), we view the collection of the elements of 2×2 Hessian matrix $d^2 f(z)$, $z \in \Omega$, as

the corresponding homogeneous polynomial in the Taylor expansion of f at point $z = (x, y)$:

$$d^2 f(z) \equiv \frac{\partial^2 f(z)}{\partial x^2} x^2 + 2\frac{\partial^2 f(z)}{\partial x \partial y} xy + \frac{\partial^2 f(z)}{\partial y^2} y^2.$$

Now, imposing the additional requirement that $\mathcal{D}$ be a uniform triangulation, we obtain that $h^2 \sim N^{-1}$, and thus,

$$\|f - f_N\|_{L_p(\Omega)} \leq CN^{-1} \|d^2 f\|_{L_p(\Omega)}. \tag{1.8}$$

This convergence rate can only be guaranteed for smooth f, and obviously not for f with discontinuities along the edges.

Isotropic triangulations. Uniform partitions do not take into account any individual properties of the function to be approximated. That is why the estimate (1.8) can be improved using adaptively generated partitions.

As in the uniform case, we do not intend to rigorously obtain or prove the estimates. We are rather looking for some arguments and ideas of how these estimates may be derived. Hence, we assume that for large values of N, $d^2 f$ can be viewed as a constant over each triangle Δ (in other words, we replace f with its approximation by quadratic function).

Similarly to (1.7), we note that the local error of approximation satisfies

$$e(f)_{L_p(\Delta)} \leq C h_\Delta^2 \|d^2 f\|_{L_p(\Delta)},$$

where $h_\Delta := \operatorname{diam}(\Delta)$. Taking into account that the triangulations $\{\mathcal{D}_N\}$ are isotropic, all of the triangles must have minimal angle bounded away from 0, and therefore $h_\Delta^2 \sim |\Delta|$, where $|\Delta|$ here denotes the area of triangle Δ. In other words, h_Δ^2 can be bounded below and above by some multiple of $|\Delta|$, for all Δ in the triangulation. If we denote by τ the positive number defined by $\dfrac{1}{\tau} := \dfrac{1}{p} + 1$, we get the following

$$e(f)_{L_p(\Delta)} \leq C \|d^2 f\|_{L_\tau(\Delta)}. \tag{1.9}$$

Assume now that we can construct adaptive isotropic triangulations $\mathcal{D}_N$ with $N := \#\mathcal{D}_N$ that equidistribute the local error in the sense that for some prescribed $\varepsilon > 0$,

$$c\varepsilon \leq e(f)_{L_p(\Delta)} \leq \varepsilon, \tag{1.10}$$

with $c > 0$ independent of Δ and N. Then, condition (1.10) immediately implies that

$$\|f - f_N\|_{L_p(\Omega)} \leq N^{1/p}\varepsilon. \tag{1.11}$$

We can estimate ε from above using inequality (1.9)

$$N(c\varepsilon)^\tau \leq \sum_{\Delta\in\mathcal{D}_N} [e(f)_{L_p(\Delta)}]^\tau \leq \sum_{\Delta\in\mathcal{D}_N} C^\tau \|d^2 f\|^\tau_{L_\tau(\Delta)} = C^\tau \|d^2 f\|^\tau_{L_\tau(\Omega)}. \tag{1.12}$$

Combining inequalities (1.11) and (1.12), we get:

$$\|f - f_N\|_{L_p(\Omega)} \leq CN^{-1}\|d^2 f\|_{L_\tau(\Omega)}. \tag{1.13}$$

Comparing with (1.8), the same rate $O(N^{-1})$ is proven for a wider class of functions, and coincides with the rate of wavelet approximation using threshold of N absolutely largest coefficients.

One of the approaches for constructing isotropic triangulations with the error equidistribution property (1.10) is so-called Newest Vertex Bisection method [1]. At each step of this algorithm, a triangle of maximal error is being bisected from the most recently created vertex. It is possible to show that the generated triangulations will satisfy isotropy property (1.6). However, such an algorithm cannot guarantee the error equidistribution exactly in the sense of (1.10), and thus does not lead to the same estimate (1.13). On the other hand, for functions $f \in B^2_{\tau,\tau}$, it is possible to show (see [2]) that

$$\|f - f_N\|_{L_p} \leq CN^{-1}|f|_{B^2_{\tau,\tau}}$$

which provides the optimal rate $O(N^{-1})$ as well.

Anisotropic triangulations. The following theoretical results provide the asymptotic rate of best piecewise-linear approximation over triangulation with N elements.

Theorem 1.4 ([24])**.** *Let $f \in C^{(2)}(\Omega)$ and $\det(d^2 f)$ be the determinant of the 2×2 Hessian matrix of f. Then*

$$\limsup_{N\to\infty} Ne_N(f)_{L_p(\Omega)} \leq C\|\sqrt{|\det(d^2 f)|}\|_{L_\tau(\Omega)}, \quad \frac{1}{\tau} = \frac{1}{p} + 1,$$

where C is a positive constant independent of N and f;

If in addition f is convex, then

$$\liminf_{N\to\infty} Ne_N(f)_{L_p(\Omega)} \geq c\|\sqrt{|\det(d^2 f)|}\|_{L_\tau(\Omega)}, \quad \frac{1}{\tau} = \frac{1}{p} + 1,$$

while c is a positive constant independent of f and N.

The essence of this theorem is that the optimal triangulations have the following properties: (a) the aspect ratio of triangles is locally adapted by the Hessian of f, and thus, optimal triangulations are isotropic with respect to a scaled metric induced by the local value of the determinant of the hessian $\sqrt{|\det(d^2 f)|}$ on each triangle (and hence may be anisotropic in the Euclidean metric sense); (b) the triangulation should equidistribute the local approximation error.

From the numerical point of view, the algorithm taking into account the properties above is executed using Delaunay triangulation. This algorithm is efficient however lacking some features:

(i) it is based on the value of $\sqrt{|\det(d^2 f)|}$, and therefore cannot be applied to non-smooth or noisy functions;

(ii) the meshes $\{\mathcal{D}_N\}_N$ produced by these algorithms are non-hierarchical in the following sense: for $M > N$, the triangulation $\mathcal{D}_M$ is not a refinement of $\mathcal{D}_N$.

Hierarchical structures in meshes allow the construction of wavelet bases and multiresolutional analysis, which plays an important role in Image Coding and Compression. There have been some greedy refinement procedures [7] that even enlarge the class of function that satisfy the estimate in Theorem 1.4, but still do not achieve estimate (1.4) for simple cartoon images.

Chapter 2

Hierarchial Adaptive Triangulations for Cartoon Images

2.1 Preliminaries and General Principle

In this chapter, we consider the problem of constructing adaptive triangulations for the approximation of the characteristic function

$$f(x) := \chi_\Omega(x), \quad x \in [0,1]^2,$$

where Ω is a subset of $[0,1]^2$ with a piecewise-smooth boundary.
Conditions on the set Ω. We will often require Ω to be convex.

- The set $\Omega \subset \mathbb{R}^2$ is called **convex** if for all $X, Y \in \Omega$, $XY \subset \Omega$. A curve γ is called **convex** if it is the boundary of some convex set.

- For a given set $S \subset \mathbb{R}^2$, the **convex hull** of S is the smallest convex set containing S. We denote the convex hull of S by $\mathrm{conv}(S)$. Alternatively, $\mathrm{conv}(S)$ may be defined as a set of all convex combinations of points of S, *i.e.*,

$$\mathrm{conv}(S) := \left\{ \sum_{k=1}^{n} \alpha_k x_k : x_k \in S,\, \alpha_k \in [0,1],\, \sum_{k=1}^{n} \alpha_k = 1,\, n \geq 1 \right\}. \qquad (2.1)$$

Triangulation structure. By $\mathcal{D}_0$ we denote the initial triangulation of $[0,1]^2$, and all the subsequent triangulations/partitions are denoted by $\mathcal{D}_i$, $i \geq 1$. Each triangulation is a set of triangles Δ.
In other words,

$$\bigcup_{\Delta \in \mathcal{D}_i} \Delta = [0,1]^2,$$

and

$$\text{int}(\Delta_1 \cap \Delta_2) = \emptyset, \quad \text{for any two distinct triangles } \Delta_1, \Delta_2 \in \mathcal{D}_i.$$

Definition 2.1. *When a triangle Δ is being split by some rule, the triangles obtained are called* **children** *of Δ, and the set of all children of Δ is denoted by $\mathcal{C}(\Delta)$.*
If $\Delta' \in \mathcal{C}(\Delta)$, then we say that Δ is the **parent** *of Δ', and write $\Delta = \mathcal{P}(\Delta')$.*
The set of children of Δ of kth generation will be denoted by $\mathcal{C}^{(k)}(\Delta)$ (for instance, $\mathcal{C}^{(1)}(\Delta) = \mathcal{C}(\Delta)$).

Error of approximation over a triangulation.

Definition 2.2. *Given $S \subset \mathbb{R}^2$, by $E_r(f,S)_p$ we denote the error of best L_p-approximation of a function $f \in L_p(S)$ by polynomials (of two variables) of total degree $< r$.*
For a partition $\mathcal{D}$ of $[0,1]^2$, we define

$$\sigma_r(f,\mathcal{D})_p^p := \sum_{\Delta \in \mathcal{D}} E_r(f,\Delta)_p^p \tag{2.2}$$

to be pth power of the global error of approximation of f by piecewise polynomials of total degree $< r$ on the partition $\mathcal{D}$.

General algorithm. The whole sequence of *hierarchial* partitions can be understood as a collection $\mathcal{F}$ of trees (or, more precisely, as a forest) where the vertices of the graph are triangles. Suppose that initial partition $\mathcal{D}_0$ consists of k_0 triangles. These triangles from $\mathcal{D}_0$ form the roots of the trees, and so we start with graph $\mathcal{F}_0$: $V(\mathcal{F}_0) = \mathcal{D}_0$ and $E(\mathcal{F}_0) = \emptyset$.

Now given a forest $\mathcal{F}_k$, we subdivide some of the triangles from $\mathcal{F}_k$ that do not have children (so-called **leaves**) using some rule. In our case, for a precision $\epsilon > 0$

given in advance, our algorithm suggests that Δ is to be subdivided iff $e(\Delta) > \epsilon$. $e(\Delta)$ is called the **local error indicator** or the **decision function**.

For each $\mathcal{F}_k$, denote this set of subdivided triangles by $\mathcal{M}_k$. Then, the forest $\mathcal{F}_{k+1}$ obtained after the subdivision of $\mathcal{F}_k$ can be described as follows:

$$V(\mathcal{F}_{k+1}) := V(\mathcal{F}_k) \cup \bigcup_{\Delta \in \mathcal{M}_k} \mathcal{C}(\Delta), \quad E(\mathcal{F}_{k+1}) := E(\mathcal{F}_k) \cup \bigcup_{\Delta \in \mathcal{M}_k} \bigcup_{\Delta' \in \mathcal{C}(\Delta)} \{\Delta\Delta'\},$$

where $\{\Delta\Delta'\}$ denotes the edge connecting the vertices Δ and Δ'. The triangulation $\mathcal{D}_{k+1}$ is the set of all the leaves in $\mathcal{F}_{k+1}$.
The procedure stops when $\mathcal{M}_N = \emptyset$, and hence the final partition $\mathcal{D}_N$ consists of all the leaves in $\mathcal{F}_N$. Note that in comparison with Section 1.5, here N indicates the length of a longest (simple) path between the leaves and the corresponding roots in the final forest of hierarchical triangulations.

Remark 2.1. *Note that in this hierarchial structure if two triangles $\Delta_1, \Delta_2 \in \mathcal{F}_N$ have a common interior point then $\Delta_1 \subset \Delta_2$ or $\Delta_2 \subset \Delta_1$, and one of the triangles is a child of some generation of the other one, i.e., one of them is obtained after a certain number of subdivisions of the other one.*

2.2 Approximation of Convex Curves

In this section, we introduce an algorithm that constructs a straight line approximation to a curve from a special class. We first present some definitions and auxiliary facts that we use in the study of properties of this approximation.

2.2.1 Centroid of Convex Figures

We start with the definition of the **moment about a line**.

Definition 2.3. *The* **moment about a line** *of a point-mass is the product of the mass with the directed distance from the point to the line.*

In the case of the line l given by the equation $px + qy + r = 0$, $|p| + |q| \neq 0$ (with either $p > 0$ or $p = 0, q > 0$, which can always be achieved by multiplying both sides

of the equation of the line by -1 if necessary), and the mass m located at the point (x_0, y_0), the moment about the line l of the point-mass m becomes

$$m\frac{px_0 + qy_0 + r}{\sqrt{p^2 + q^2}}.$$

This concept can be easily generalized to a moment of an area (plane figure) about a line via integrals. For instance, the moments of the area bounded by the curves $y = f(x)$, $y = 0$, $x = a$, $x = b$ ($f(x) \geq 0$, $x \in [a, b]$), about the x- and y-axis are defined as follows:

$$M_x(A) := \int_a^b \frac{(f(x))^2}{2}\, dx, \quad M_y(A) := \int_a^b xf(x)\, dx.$$

Definition 2.4. *The* **centroid** *of a plane figure* Φ *is the intersection of all straight lines about which* Φ *has zero moment.*

The centroid can be understood as the arithmetic mean of all points within the figure with equally distributed mass. There are various practical methods of finding it such as the plumb line method, the balancing method (for convex figures), etc. In the case of a plane figure in the first quadrant bounded by the x-axis, the graph of the function $y = f(x)$, and lines $x = a$ and $x = b$, the coordinates of the centroid M can be found analytically via the formula:

$$x_M = \frac{\int_a^b xf(x)\, dx}{\int_a^b f(x)\, dx}, \qquad y_M = \frac{\frac{1}{2}\int_a^b (f(x))^2\, dx}{\int_a^b f(x)\, dx}.$$

Lemma 2.1 ([16, pp. 248-251],[20, pp. 499-503],[33, pp. 48-50])**. Elementary properties of centroids**

(a) The centroid of a finite collection of n *points* $X_1, X_2, \ldots, X_n$ *is the point*

$$M = \frac{1}{n}\sum_{i=1}^{n} X_i.$$

(b) **Decomposition Rule:** *Suppose that a plane figure* Φ *is divided into a finite number of pairwise essentially disjoint figures* $\Psi_1, \Psi_2, \ldots, \Psi_n$ *with areas*

$A_1, A_2, \ldots, A_n$ and centroids $M_1, M_2, \ldots, M_n$. Then the centroid M of the figure Φ can be computed as follows:

$$M = \frac{\sum_{i=1}^{n} M_i A_i}{\sum_{i=1}^{n} A_i}.$$

(c) The centroid of a triangle with vertices A, B, C is the point of intersection of all three medians, and can be found as follows

$$M = \frac{A + B + C}{3}.$$

(d) Let P be a non-self-intersecting closed polygon with n vertices $P_1, P_2, \ldots, P_n$. Then the centroid of the figure bounded by the polygon P is the point

$$M = \frac{\sum_{i=0}^{n-1} (P_i + P_{i+1}) A_i}{3 \sum_{i=0}^{n-1} A_i},$$

where $A_i := \det[\overrightarrow{OP_i}; \overrightarrow{OP_{i+1}}]$ is the (signed) area of the parallelogram spanned by the vectors $\overrightarrow{OP_i}$ and $\overrightarrow{OP_{i+1}}$, O is the origin and $P_0 := P_n$.

(e) If a plane figure has a line of symmetry, then its centroid lies on this line.

(f) The centroid of a convex figure always lies inside the figure (more precisely, in its interior for the figures with non-empty interior in $\mathbb{R}^2$).

(g) Suppose that $\Phi \subset \Psi$, and Ψ is a convex figure in the plane. Then the centroid of Φ lies inside Ψ.

(h) Let Φ be a set entirely lying in a half-plane about a straight line l. Then the centroid of Φ lies in the same half-plane about the line l (more precisely, in the interior of the half-plane for non-degenerate cases).

(i) Let Φ be a planar set that is the union $\Phi = \Phi_1 \cup \Phi_2$ of two sets Φ_1 and Φ_2 with non-empty interiors and $\lambda_2(\Phi_1 \cap \Phi_2) = 0$. Then the centroids of the three sets Φ_1, Φ_2 and Φ are collinear (i.e., they lie on the same line).

(j) The moment of a plane figure Φ about any line passing through its centroid is zero.

(k) A translation of a plane figure by a vector translates its centroid by the same vector.

It is worth noting that part (h) follows from part (g) since a half-plane is a convex set, and part (i) follows from part (b). All other parts follow from the definition of the centroid and the properties of the moment of the area about a line.
In our setting, we are mostly dealing with convex objects. In order to study certain features of our algorithm presented in the next subsection, we need some further properties of the centroid of convex figures.
The following classical result was proved by Winternitz and first published in 1923 in [3, p. 54]. This theorem remained unnoticed for some period of time and was rediscovered by Lavrent'ev and Lusternik [21, pp. 357-358] in 1935, Neumann [25, Theorem 2.5, p. 229] in 1945, Yaglom and Boltyanski [33, p. 50] in 1951, Ehrhart [12, pp. 483-485] in 1955 and Newman [26, Abstract, p. 510] in 1958.

Lemma 2.2 (**Winternitz Theorem**). *If a convex figure is divided into two parts by a line l that passes through its centroid, then the ratio of the areas of the two parts always lies between the bounds $4/5$ and $5/4$.*

Proof. One of the proofs of the Winternitz theorem can be found in [33, pp. 196-198]. It is based on a construction of a triangle that has a smaller ratio of the areas of the parts cut by the same line, and then on further analysis of this ratio for the case when the figure is a triangle. As a conclusion, it is proven that *the ratio of the areas S_1 and S_2 of two figures obtained by splitting a convex set Φ with a straight line l passing through the centroid of Φ is minimal if and only if Φ is a triangle and the line l is parallel to one of the sides of this triangle.* This in particular implies that

for a non-triangular convex figure the ratio S_1/S_2 lies even in a more narrow interval than $[4/5, 5/4]$. □

Remark 2.2. *Note that without the condition of convexity one can only guarantee the existence of a point O inside the figure such that any line crossing the figure divides it into two figures with areas S_1 and S_2 and $1/2 \leq S_1/S_2 \leq 2$ (the proof of this fact uses Helly's well-known theorem and can be found in* [33, pp. 156-157] *as well).*

2.2.2 The Main Idea of Subdivision

Let Δ be a triangle such that $\Delta \cap \partial\Omega$ is a segment of a straight line. The main idea of the algorithm we propose is based on the possibility of the full reconstruction of $\Delta \cap \partial\Omega$ in this case, and uses only the knowledge about the coordinates of the centroid of $\Delta \cap \Omega$ or $\overline{\Delta \backslash \Omega}$ (whichever is a triangle).
Indeed, given $\Delta := \triangle ABC$ and a point M inside it (later on, we will choose M to be the centroid of $\Delta \cap \Omega$ or $\overline{\Delta \backslash \Omega}$), the following procedure leads to the construction of the unique $\triangle ADE$ such that $D \in \overrightarrow{(A, B)}$, $E \in \overrightarrow{(A, C)}$ and M is the centroid of $\triangle ADE$ (see Figure 2.1):

Construction 2.1. (Main Idea of Subdivision)
Step 1. Construct the point M' such that $\overrightarrow{AM'} = 3/2\overrightarrow{AM}$.

Step 2. Draw the lines l_1 and l_2 passing through M' and parallel to AC and AB, respectively.

Step 3. Construct points D and E such that $\overrightarrow{AD} = 2\overrightarrow{AK}$ and $\overrightarrow{AE} = 2\overrightarrow{AL}$, where K is the point of intersection of l_1 and $\overrightarrow{(A, B)}$, and L is the point of intersection of l_2 and $\overrightarrow{(A, C)}$.

In this construction,

$$\overrightarrow{AM'} = \overrightarrow{AK} + \overrightarrow{KM'} = \overrightarrow{AK} + \overrightarrow{AL} = \frac{1}{2}\left(\overrightarrow{AD} + \overrightarrow{AE}\right),$$

and hence, the points M', K and L are the midpoints of the sides DE, AD and AE, respectively, and so the centroid of $\triangle ADE$ is located at M.

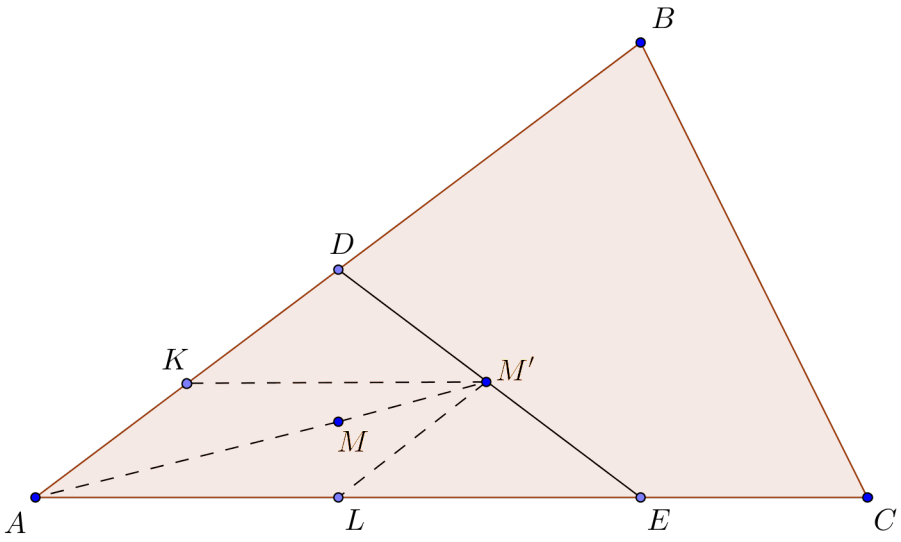


Figure 2.1: Construction of $\triangle ADE$ using the location of its centroid

Conclusion: Thus, if $\Delta \cap \partial\Omega$ is a segment of a straight line, Construction 2.1 will fully reconstruct it. In the general case when $\Delta \cap \partial\Omega$ is a connected curve but not necessarily a straight line segment (and M is the centroid of $\Delta \cap \Omega$ or $\overline{\Delta \backslash \Omega}$), the same procedure produces the segment DE which is some approximation to this curve. Note that one of the points D or E may not be in Δ and can lie on the extension of one of the sides of the triangle (see Figure 2.2 below for an example; in the following subsections, we will show that only one of the points D or E may be outside Δ).

Remark 2.3. *The construction 2.1 can be expressed analytically as follows:*
Input points: A, B, C, M.
Step 1 (preliminary): Verify that the points A, B, C *are not collinear, and that the point* M *lies in the interior of* $\triangle ABC$ *(this can be done by verifying that each of the three triangles* ABM, ACM *and* BCM *has a positive area, and that the sum of these three areas is equal to the area of* $\triangle ABC$*).*
Step 2: Solve the system of linear equations with two variables t *and* s

$$3(M - A) = s(B - A) + t(C - A),$$

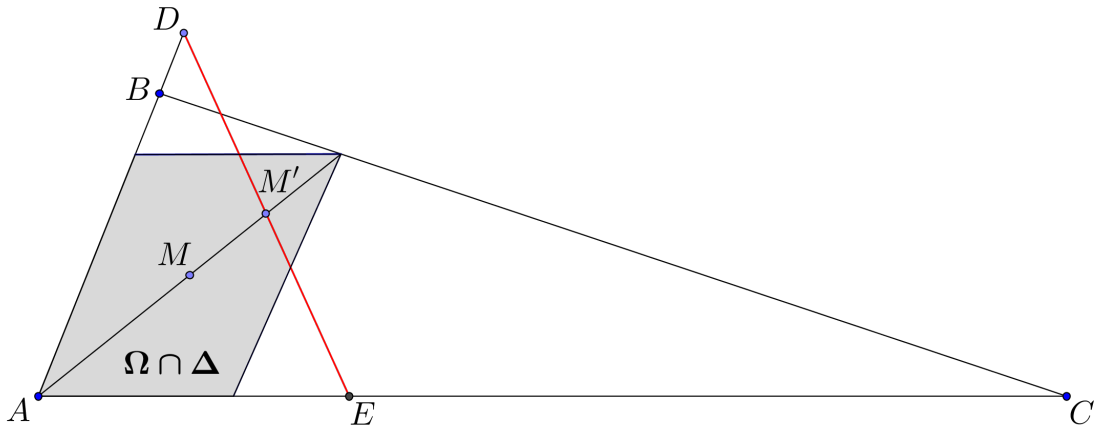


Figure 2.2: Point D lies outside $\triangle ABC$

(which can be derived from the requirement that

$$\begin{cases} D = A + s(B - A), \\ E = A + t(C - A), \\ M = (A + D + E)/3, \end{cases} ,$$

and has a unique solution since $\triangle ABC$ has non-empty interior, and the matrix $[B - A, C - A]$ is therefore invertible).
Output points: D, E.

2.2.3 Centroids of Angular Sets

In this subsection, we restrict our attention to specific types of plane figures, namely sets enclosed by the sides of an angle and some curve crossing the angle (here and throughout the thesis, such sets are closed with the enclosing set always included in the figure described). We establish results about the centroids of such sets and the analysis of the application of Construction 2.1 to these 'angular sets'. We start with a precise definition of these objects:

Definition 2.5. *Let $\angle A$ be the geometric figure on the Cartesian plane consisting of two different rays p_1 and p_2 originating at a common vertex A such that p_1 and p_2*

do not lie on the same line (i.e., $\angle A$ is not a ray or a line).

- *By $S(\angle A)$, we denote the convex hull of $\angle A$, i.e., the set of points inside the angle together with the sides of the angle. Note that $S(\angle A)$ is always a region enclosed by an angle with degree measure in the interval $(0°, 180°)$.*

- *By $\Gamma(\angle A)$, we denote the set of simple rectifiable curves $\gamma = r(t)$, $t \in [a, b]$, (i.e., curves of finite length that have no points of self-intersection except possibly $r(a) = r(b)$) such that*

$$\forall t \in (a, b): \qquad r(t) \in S(\angle A) \backslash \angle A \quad \text{and} \quad r(a) \in p_1, r(b) \in p_2.$$

  Note that one or both points $r(a)$ and $r(b)$ may coincide with the vertex A.

- *For each $\gamma \in \Gamma(\angle A)$, by $\Phi_{\angle A}(\gamma)$ we denote the (closed) set enclosed by the curve γ and the sides of $\angle A$ (the 'angular set').*

We also introduce some auxiliary notations:

Definition 2.6. *Given two points P and Q on a simple curve γ,*

- *$\gamma(P, Q)$ denotes the portion of the curve between the points P and Q;*

If, in addition, $PQ \cap \gamma = \{P, Q\}$, or in other words, $PQ \cup \gamma(P, Q)$ remains to be a simple curve (which will be the case if γ is convex for instance)

- *$Seg(\gamma, P, Q)$ denotes the closed set of the points enclosed by $\gamma(P, Q)$ and the straight line segment PQ.*

  Note that $Seg(\gamma, P, Q)$ is a closed set with $\partial Seg(\gamma, P, Q) = \gamma(P, Q) \cup PQ$.

Now we are ready to present some facts about the centroids of 'angular sets'.

Lemma 2.3. *Let $\angle A$ be an angle on the plane and $\gamma \in \Gamma(\angle A)$ be a convex curve intersecting the sides of the angle at the points P and Q (the set $Seg(\gamma, P, Q)$ is convex). Let, in addition, $\mathcal{R}$ be a set that is homothetic to $Seg(\gamma, P, Q)$ with ratio $2:3$ about point A, i.e.,*

$$\mathcal{R} := \{X \in S(\angle A) : \overrightarrow{AX} = 2/3\overrightarrow{AY}, \text{ for some } Y \in Seg(\gamma, P, Q)\}. \tag{2.3}$$

Then, the centroid of $\Phi_{\angle A}(\gamma)$ lies in the set $\mathcal{R}$.

Remark 2.4. *The set $\mathcal{R}$ can be described in a simpler form as*

$$\mathcal{R} = \frac{1}{3}A + \frac{2}{3}Seg(\gamma, P, Q).$$

Proof. Since the curve γ is convex, we can split the set $\Phi_{\angle A}(\gamma)$ into n curvilinear triangles by drawing $n-1$ rays originating at A and dividing $\angle A$ into n equal angles. Denote by $H_1, \ldots, H_{n-1}$ the points of intersection of the constructed rays with γ, and, for simplicity of notation, let $H_0 := P$ and $H_n := Q$. Let M and M_n be the centroids of $\Phi_{\angle A}(\gamma)$ and the polygon $AH_0H_1 \ldots H_n$, respectively. For any $\varepsilon > 0$, we can choose a sufficiently large n such that $|MM_n| < \varepsilon$. Note that, for each $1 \le i \le n$, $H_{i-1}H_i \subset Seg(\gamma, P, Q)$ due to the convexity of $Seg(\gamma, P, Q)$. In particular, the midpoint of $H_{i-1}H_i$ lies within $Seg(\gamma, P, Q)$. Then by Lemma 2.1(c), the centroid of $\triangle AH_{i-1}H_i$ lies in the set $\mathcal{R}$ defined in (2.3). By Lemma 2.1(b), M_n belongs to the convex hull of the centroids of $\triangle AH_{i-1}H_i$'s, and thus, since $\mathcal{R}$ is convex, $M_n \in \mathcal{R}$. This implies that $\mathrm{dist}(M, \mathcal{R}) < \varepsilon$. Since ε was an arbitrary positive value, we conclude that $\mathrm{dist}(M, \mathcal{R}) = 0$, and therefore, because $\mathcal{R}$ is a closed set, $M \in \mathcal{R}$. □

Remark 2.5. *A curve $\gamma \in \Gamma(\angle A)$ is convex if and only if one of the sets $\Phi_{\angle A}(\gamma)$ or $\overline{S(\angle A) \backslash \Phi_{\angle A}(\gamma)}$ is convex.*

Remark 2.6. *Note that the statement of Lemma 2.3 and its proof remain valid if one or both of the points P and Q coincide with A. The only adjustment required in the proof is to take into account that, in this case, not necessarily all of the $n-1$ rays originating at A and dividing $\angle A$ into n equal angles intersect the curve γ. However, for a sufficiently large n, at least two of the $n-1$ rays intersect γ; we denote by $H_k, \ldots, H_m$, $1 \le k < m \le n-1$, the respective points of intersection. Then one can consider the union of $\triangle AH_{i-1}H_i$'s, $k+1 \le i \le m$, as an approximation of the set $\Phi_{\angle A}(\gamma)$.*

Remark 2.7. *Constant $2/3$ in (2.3) cannot be replaced with any other constant. This can be easily seen by looking at the case when γ is a straight line segment.*

Next, we want to look at the centroids of two different angular sets simultaneously. Namely, the question we are interested in is as follows: given two curves $\gamma_1, \gamma_2 \in$

$\Gamma(\angle A)$ with no points of intersection, is it possible that the centroids of $\Phi_{\angle A}(\gamma_1)$ and $\Phi_{\angle A}(\gamma_2)$ coincide? It turns out that, under some conditions on γ_1 and γ_2, this is not possible.

Lemma 2.4. *Let $\gamma_1, \gamma_2 \in \Gamma(\angle A)$ be two convex curves that can be separated by some straight line l not intersecting any of the two curves (i.e., γ_1 and γ_2 lie in the interior of different half-planes about l). Then the centroids of $\Phi_{\angle A}(\gamma_1)$ and $\Phi_{\angle A}(\gamma_2)$ are distinct.*

Note that the assumptions of the lemma imply that γ_1 and γ_2 do not have points of intersection.

Proof. Let $\{P_1, Q_1\} = \gamma_1 \cap \angle A$ and $\{P_2, Q_2\} = \gamma_2 \cap \angle A$ (it is possible that, not simultaneously, P_1 coincides with Q_1 or P_2 coincides with Q_2). Also, let $\mathcal{R}_1$ and $\mathcal{R}_2$ be the sets defined via equation (2.3) for the curves γ_1 and γ_2, respectively. Since γ_1 and γ_2 are convex and separated by the line l that does not intersect any of the two curves, the convex hulls of γ_1 and γ_2, *i.e.*, sets $Seg(\gamma_1, P_1, Q_1)$ and $Seg(\gamma_2, P_2, Q_2)$, lie in the interior of different half-planes about the line l, and thus, $Seg(\gamma_1, P_1, Q_1) \cap Seg(\gamma_2, P_2, Q_2) = \emptyset$. This immediately implies that

$$\mathcal{R}_1 \cap \mathcal{R}_2 = \emptyset, \tag{2.4}$$

whereas $\mathcal{R}_1$ and $\mathcal{R}_2$ are homothetic (with the same ratio of homothety) to $Seg(\gamma_1, P_1, Q_1)$ and $Seg(\gamma_2, P_2, Q_2)$, respectively. Therefore, taking into account Lemma 2.3, we can now conclude that the centroids of $\Phi_{\angle A}(\gamma_1)$ and $\Phi_{\angle A}(\gamma_2)$ are distinct. □

Remark 2.8. *The condition of convexity imposed on the curves γ_i, $i = 1, 2$, cannot be removed from the statement of Lemma 2.4. The next example shows that even if only one of the curves is not convex, the statement of Lemma 2.4 is no longer valid. Indeed, we present a sketch of such an example in Figure 2.3.*
Curve γ_1 is a straight line segment P_1Q_1 while curve γ_2 consists of a part of a fixed $\triangle KLN$ and two parallel lines crossing the sides of the angle at points P_2 and Q_2. Both of the curves are symmetric about the bisector of $\triangle AP_1Q_1$ towards the side P_1Q_1, and thus the centroids of $\Phi_{\angle A}(\gamma_1)$ and $\Phi_{\angle A}(\gamma_2)$ lie on that bisector. We choose

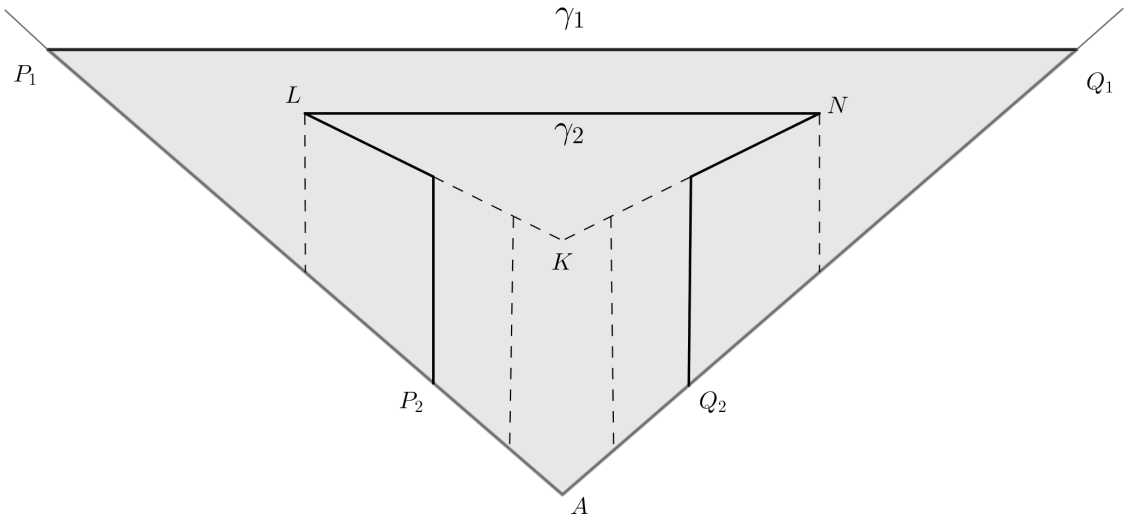


Figure 2.3: Non-convex and convex curve separable by a line

$\triangle KLN$ so that its centroid is located above the centroid of $\triangle AP_1Q_1$. Hence, if the distance P_2Q_2 is small enough, the centroid of $\Phi_{\angle A}(\gamma_2)$ is located above the centroid of $\Phi_{\angle A}(\gamma_1)$. If P_2Q_2 is large enough (so that γ_2 becomes convex), then the centroid of $\Phi_{\angle A}(\gamma_1)$ is located above the centroid of $\Phi_{\angle A}(\gamma_2)$ by Lemma 2.4. By continuity argument, one can conclude that for some intermediate length of P_2Q_2, the centroids of $\Phi_{\angle A}(\gamma_1)$ and $\Phi_{\angle A}(\gamma_2)$ will coincide.
For instance, one can choose the points with the following coordinates in the Cartesian plane:

$$A(0,0),\ P_1(-9.8325, 9.8325),\ Q_1(9.8325, 9.8325),\ K(0,5),$$
$$L(-3,8),\ N(3,8),\ P_2(-0.1, 0.1),\ Q_2(0.1, 0.1),$$

and in this case the centroids of $\Phi_{\angle A}(\gamma_1)$ and $\Phi_{\angle A}(\gamma_2)$ will be both located at $M(0, 6.555)$.

The condition of separation of two convex curves by a line is quite strong. One might think that it can be replaced with a weaker condition on γ_1 and γ_2 that one of the sets of $\Phi_{\angle A}(\gamma_1)$ and $\Phi_{\angle A}(\gamma_2)$ is a subset of the other one (assuming again that both curves are convex). However, the following example shows that for some polygonal curves, the latter condition does not guarantee that the centroids of $\Phi_{\angle A}(\gamma_1)$ and $\Phi_{\angle A}(\gamma_2)$ are distinct.

Example 2.1 (Figure 2.4)**.** Given an $\angle A$, we choose γ_1 to be a straight line segment PQ. We choose γ_2 to be the boundary of $\triangle AKL$ such that the points K, L lie on PQ, points A, P, K, L, Q are in clockwise order, and $|KP| = |LQ|$. $|KP| = |LQ|$ implies that the midpoints of KL and PQ coincide, and thus, it is easy to see that the sets $\Phi_{\angle A}(\gamma_1) = \triangle APQ$ and $\Phi_{\angle A}(\gamma_2) = \triangle AKL$ have their centroids located at the same point.

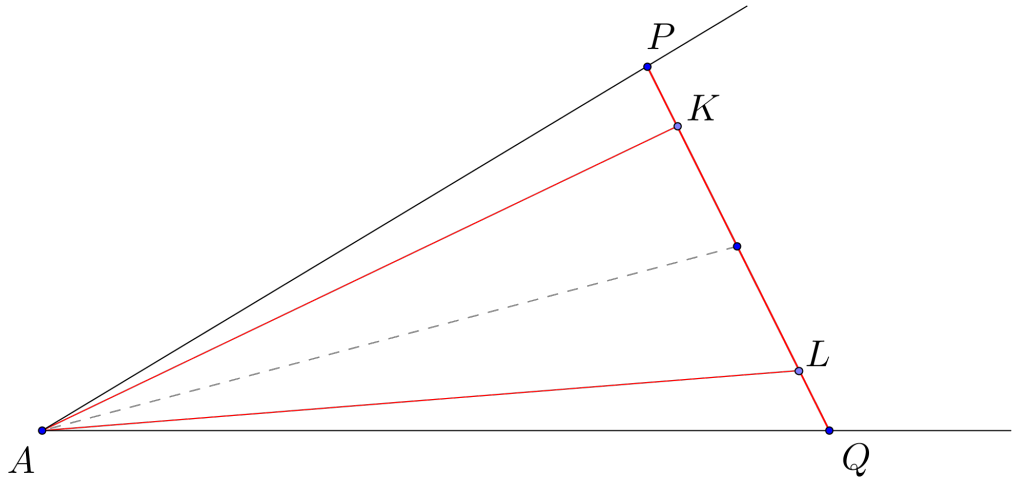


Figure 2.4: Centroids of $\triangle APQ$ and $\triangle AKL$ coincide

In the following subsections, we construct a linear approximation for any convex $\gamma \in \Gamma(\angle A)$ and some fixed angle $\angle A$. That is why, in our further discussion, we focus on the case where γ_2 is a straight line segment. It turns out that, in this case, the condition that either $\Phi_{\angle A}(\gamma_1) \subset \Phi_{\angle A}(\gamma_2)$ or $\Phi_{\angle A}(\gamma_2) \subset \Phi_{\angle A}(\gamma_1)$ can be sufficient to establish results similar to Lemma 2.4. More precisely, in the following Lemmas 2.5 and 2.6, we prove that, if $\Phi_{\angle A}(\gamma_1) \subset \Phi_{\angle A}(\gamma_2)$ or $\Phi_{\angle A}(\gamma_2) \subset \Phi_{\angle A}(\gamma_1)$, the centroids of $\Phi_{\angle A}(\gamma_1)$ and $\Phi_{\angle A}(\gamma_2)$ can coincide only in the case described in Example 2.1.

Lemma 2.5. *Let $\angle A$ be some angle, $\gamma \in \Gamma(\angle A)$ be convex, K and L be two points lying on the different sides of $\angle A$ such that $\triangle AKL \subsetneq \Phi_{\angle A}(\gamma)$. Then the centroids of $\triangle AKL$ and $\Phi_{\angle A}(\gamma)$ cannot coincide.*

Proof. Let γ be a convex curve crossing $\angle A$, and KL be a segment with the endpoints on the sides of $\angle A$ such that $\triangle AKL \subsetneq \Phi_{\angle A}(\gamma)$.

By way of contradiction, assume that the centroids of $\triangle AKL$ and $\Phi_{\angle A}(\gamma)$ coincide. Note that

$$\Phi_{\angle A}(\gamma) = \triangle AKL \cup \overline{(\Phi_{\angle A}(\gamma) \backslash \triangle AKL)}, \quad \text{int}\left(\triangle AKL \cap \overline{(\Phi_{\angle A}(\gamma) \backslash \triangle AKL}\right) = \emptyset,$$

and since the centroids of $\triangle AKL$ and $\Phi_{\angle A}(\gamma)$ coincide, by Lemma 2.1(b), the centroids of $\overline{\Phi_{\angle A}(\gamma) \backslash \triangle AKL}$ and $\triangle AKL$ coincide as well. However, the sets $\overline{\Phi_{\angle A}(\gamma) \backslash \triangle AKL}$ and $\triangle AKL$ lie in different half-planes about the line $\overleftrightarrow{KL}$, and thus, by Lemma 2.1(h), the centroids of $\overline{\Phi_{\angle A}(\gamma) \backslash \triangle AKL}$ and $\triangle AKL$ are distinct. We arrived at a contradiction. Hence, our assumption is false, and the centroids of $\triangle AKL$ and $\Phi_{\angle A}(\gamma)$ are not the same. □

Lemma 2.6. *Let $\angle A$ be some angle, $\gamma \in \Gamma(\angle A)$ be convex, K and L be two points lying on the different sides of $\angle A$. Suppose that $\Phi_{\angle A}(\gamma) \subsetneq \triangle AKL$. Then the centroids of $\Phi_{\angle A}(\gamma)$ and $\triangle AKL$ coincide if and only if γ is the boundary of some $\triangle AXY$ with $XY \subsetneq KL$, and the midpoints of XY and KL coincide.*

Proof. The sufficiency part follows from Example 2.1, and thus we need to show only the necessity part.

Assume that γ is a convex curve crossing $\angle A$, and KL is a straight line segment with the endpoints on the sides of the angle such that $\Phi_{\angle A}(\gamma)$ is a proper subset of $\triangle AKL$. Also assume that the centroids of $\Phi_{\angle A}(\gamma)$ and $\triangle AKL$ coincide. We will prove the necessity part of the lemma in two steps.

Step 1. We now show that the set $KL \cap \gamma$ contains the midpoint of KL and at least one more point distinct from the endpoints K and L.

Let N be the midpoint of KL, and define sets Φ_1 and Φ_2 as follows:

$$\Phi_1 := \Phi_{\angle A}(\gamma) \cap \triangle AKN, \qquad \Phi_2 := \Phi_{\angle A}(\gamma) \cap \triangle ALN.$$

Note that both Φ_1 and Φ_2 have non-empty interiors since otherwise $\Phi_{\angle A}(\gamma)$ would entirely lie in a single half-plane about the line $\overleftrightarrow{AN}$, and that, by Lemma 2.1(h), would be a contradiction to the fact that the centroid of $\Phi_{\angle A}(\gamma)$ (we denote this point by M) lies on the median AN of $\triangle AKL$. We can also conclude that γ and

AN have one point of intersection distinct from A. We denote this point by O and also denote by M_1 and M_2 the centroids of Φ_1 and Φ_2, respectively.
We now represent γ as

$$\gamma = \gamma_1 \cup \gamma_2,$$

where $\gamma_1 = \gamma(P, O)$, $\gamma_2 = \gamma(O, Q)$, and $P \in AK$, $Q \in AL$ are the points of intersection of γ and the sides of $\angle A$ (one or both of P and Q can coincide with the vertex A). The sets Φ_1 and Φ_2 can be now written as

$$\Phi_1 = \Phi_{\angle KAN}(\gamma_1), \qquad \Phi_2 = \Phi_{\angle LAN}(\gamma_2).$$

Lemma 2.3 implies that $M_1 \in \mathcal{R}_1$ and $M_2 \in \mathcal{R}_2$, where $\mathcal{R}_1$ and $\mathcal{R}_2$ are the sets defined via formula (2.3) for γ_1 and γ_2, respectively. Note that

$$M_1 \notin (AK \cup AN), \qquad M_2 \notin (AN \cup AL) \tag{2.5}$$

due to Lemma 2.1(h). At the same time, since $\Phi_{\angle A}(\gamma) \subset \triangle AKL$, we have the inclusion $\gamma \subset \triangle AKL$, and thus $Seg(\gamma_1, P, O)$ and $Seg(\gamma_2, O, Q)$ are subsets of $\triangle AKL$. This immediately implies that both sets $\mathcal{R}_1$ and $\mathcal{R}_2$ are subsets of $\triangle AK'L'$, where $\overrightarrow{AK'} := 2/3\overrightarrow{AK}$ and $\overrightarrow{AL'} := 2/3\overrightarrow{AL}$. On the other hand, according to Lemma 2.1(i), $M \in (M_1M_2 \cap K'L')$, which implies that

$$M_i \in K'L' \cap \partial\mathcal{R}_i, \quad i = 1, 2.$$

($M_i \in \partial\mathcal{R}_i$, since otherwise, M_i has to be an interior point of $\mathcal{R}_i$, and hence, an interior point of $\triangle AK'L'$. Obviously, due to Lemma 2.1(b), M cannot coincide with either of M_i's, and therefore, if one of the M_i's is an interior point of $\triangle AK'L'$, the point M has to be an interior point of $\triangle AK'L'$ as well which contradicts the fact that $M \in K'L'$.)

For each i, we construct the point $\widetilde{M}_i$ satisfying $\overrightarrow{A\widetilde{M}_i} = 3/2\overrightarrow{AM}_i$. Then clearly,

$$\widetilde{M}_1 \in KL \cap \partial\left(Seg(\gamma, P, O)\right), \quad \widetilde{M}_2 \in KL \cap \partial\left(Seg(\gamma, O, Q)\right),$$

and neither $\widetilde{M}_1$ nor $\widetilde{M}_2$ falls into the set of points $\{K, N, L\}$.

Taking into account that $\partial Seg(\gamma, P, O) = PO \cup \gamma_1$ and $\partial Seg(\gamma, O, Q) = OQ \cup \gamma_2$, we claim that both $\widetilde{M}_1$ and $\widetilde{M}_2$ lie on γ. Indeed, if we assume that, for instance,

$\widetilde{M}_1 \notin \gamma$, then $\widetilde{M}_1 \in PO$. However, because of condition (2.5) and the fact that $\widetilde{M}_1 \in KL$, this may only happen if $P = K$ and $O = N$, which implies that $KN \subset \gamma$, and in particular, $\widetilde{M}_1 \in KN \subset \gamma$. This contradiction proves our claim.

Therefore, both points $\widetilde{M}_1$ and $\widetilde{M}_2$ along with the segment $\widetilde{M}_1\widetilde{M}_2$ are in γ. This also implies that $N \in \gamma$ as points $\widetilde{M}_1$ and $\widetilde{M}_2$ lie in different half-planes about the line $\overleftrightarrow{AN}$. The proof of step 1 is now complete.

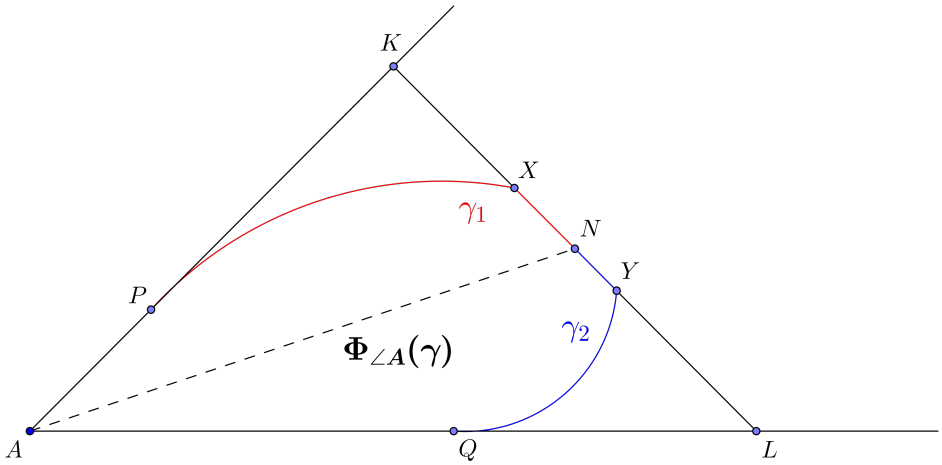


Figure 2.5: $\Phi_{\angle A}(\gamma) \subsetneq \triangle AKL$

Step 2. We now prove that γ must be the boundary of some triangle as indicated in the lemma.

Note that since $\#(\gamma \cap KL) > 2$, we may conclude that $\Phi_{\angle A}(\gamma)$ is convex, while $\overline{S(\angle A)\backslash\Phi_{\angle A}(\gamma)}$ is not. Thus, $\Phi_{\angle A}(\gamma)\cap KL$ must be some closed segment XY because the intersection of any closed convex sets is also closed and convex. It is evident that $X \neq Y$ because $\#(\gamma\cap KL) > 2$. As well, $XY \neq KL$, since otherwise, γ must coincide with segment KL and thus $\Phi_{\angle A}(\gamma) = \triangle AKL$, which contradicts $\Phi_{\angle A}(\gamma) \subsetneq \triangle AKL$. Furthermore, since $N \in XY$, we can choose X and Y to be in the same half-planes (about the line $\overleftrightarrow{AN}$) with points K and L, respectively. We now show that $\gamma = \partial\,(\triangle AXY)$.

Consider two sets

$$\Psi_1 := \Phi_{\angle A}(\gamma) \cap \triangle AXK, \qquad \Psi_2 := \Phi_{\angle A}(\gamma) \cap \triangle AYL.$$

If at least one of the Ψ_i's has a non-empty interior, then by Lemmas 2.1(h) and 2.3, its centroid is located strictly inside $\triangle AK'L'$ (and not on the boundary of the triangle), and by Lemma 2.1(i) again, the centroid of $\Phi_{\angle A}(\gamma)$ cannot lie on $K'L'$. This contradicts the fact that M is the midpoint of $K'L'$.

Therefore, both sets $\text{int}(\Psi_1)$ and $\text{int}(\Psi_2)$ are empty, and hence, $\Phi_{\angle A}(\gamma) = \triangle AXY$. Since the centroids of $\Phi_{\angle A}(\gamma)$ and $\triangle AKL$ coincide, the midpoints of XY and KL must coincide as well.
The proof of Lemma 2.6 is complete. □

This concludes the section on auxiliary statements about properties of the centroids of the angular regions, and we now move to the actual approximation of the curves that cross angles.

2.2.4 Approximation of "curves crossing angles"

Construction 2.2. *Given an angle $\angle A$ and a curve $\gamma \in \Gamma(\angle A)$, we construct the following approximation of γ:*

- *We find the centroid M of $\Phi_{\angle A}(\gamma)$.*
- *Use Construction 2.1 (page 35) to obtain $\triangle ADE$ such that D, E lie on the sides of $\angle A$ and the centroid of $\triangle ADE$ is located at M.*

Notation 2.1. *Throughout this section,*

$$DE := DE(\angle A, \gamma) \tag{2.6}$$

*is the **straight line segment approximation of the curve** γ constructed above.*

First, we want to know how many points the intersection of γ and DE has. We can establish the following result for convex γ.

Lemma 2.7. *For a given $\angle A$, let $\gamma \in \Gamma(\angle A)$ be a convex curve that is neither a straight line segment nor a boundary of some $\triangle AXY$ in $S(\angle A)$ such that the*

midpoints of the segments XY and $\overleftrightarrow{XY} \cap S(\angle A)$ coincide. If $DE = DE(\angle A, \gamma)$ is the straight line segment approximation of the curve γ, then

$$\#(\gamma \cap DE) = 2, \tag{2.7}$$

and furthermore, both points of intersection are distinct from D and E.

Proof. Since γ is convex, the number of points of intersection of γ with a straight line segment can be equal to $0, 1, 2$ or can be infinite. However, either of the equalities $\#(\gamma \cap DE) = 0, \infty$ or $\gamma \cap DE = \{D, E\}, \{D\}, \{E\}$ (together with the convexity of γ) implies that one of the three following cases takes place:

$$(1)\ \Phi_{\angle A}(\gamma) \subsetneq \triangle ADE, \quad (2)\ \triangle ADE \subsetneq \Phi_{\angle A}(\gamma), \quad (3)\ \Phi_{\angle A}(\gamma) = \triangle ADE.$$

The cases (1) and (2) can be eliminated by Lemmas 2.5, 2.6 and the fact that γ is not the boundary of some triangle inside $\angle A$. Case (3) would imply that $\gamma = DE$, and can be eliminated as well since γ is not a straight line segment by assumption. Therefore, the only possible cases are $\#(\gamma \cap DE) = 1$ or 2 and $\gamma \cap DE$ must contain at least one point distinct from D and E. In other words, we can further distinguish three cases:

$$(1')\, \gamma \cap DE = X, X \notin \{D, E\}; \quad (2')\, \gamma \cap DE = \{X, Y\},\ X \notin \{D, E\}, Y \in \{D, E\};$$
$$(3')\, \gamma \cap DE = \{X, Y\},\ X, Y \notin \{D, E\}.$$

We now show by contradiction that cases $(1')$ or $(2')$ cannot take place. Assume that DE intersects γ at a point X, $X \notin \{D, E\}$, and possibly at D or E. Due to the convexity of γ, this would mean that one of the points D, E is in $\Phi_{\angle A}(\gamma)$, and the other point is in $\overline{S(\angle A) \backslash \Phi_{\angle A}(\gamma)}$. Without loss of generality, assume that $D \in \Phi_{\angle A}(\gamma)$ and $E \in \overline{S(\angle A) \backslash \Phi_{\angle A}(\gamma)}$.
Let

$$\Phi_1 := \triangle ADE \cap \Phi_{\angle A}(\gamma), \quad \Phi_2 := \overline{\triangle ADE \backslash \Phi_1}, \quad \Phi_3 := \overline{\Phi_{\angle A}(\gamma) \backslash \triangle ADE},$$

and let M_1, M_2, M_3, M be the centroids of $\Phi_1, \Phi_2, \Phi_3, \triangle ADE$, respectively.

Since $\Phi_{\angle A}(\gamma) = \Phi_1 \cup \Phi_3$ and $\text{int}(\Phi_1 \cap \Phi_3) = \emptyset$, by Lemma 2.1(i) the centroids M_1, M and M_3 lie on the same line l^*. Furthermore, since Φ_1 and Φ_3 lie in different half-planes about the line $\overleftrightarrow{DE}$, $M_1 \neq M_3$ by Lemma 2.1(h). In addition, the Decomposition Rule (Lemma 2.1(b)) implies that the point M cannot coincide with either of M_1 or M_3. Therefore, M_1, M_3 and M are three distinct points lying on the line l^*, and $M \in M_1M_3$.

Since $\triangle ADE = \Phi_1 \cup \Phi_2$ and $\text{int}(\Phi_1 \cap \Phi_2) = \emptyset$, by Lemma 2.1(i) the points M, M_1 and M_2 lie on the same line, which clearly has to be the line l^*. Since $M \neq M_1$, the Decomposition Rule (Lemma 2.1(b)) implies that M_2 cannot coincide with either of M or M_1, and $M \in M_1M_2$. In addition, the sets Φ_2 and Φ_3 lie in different half-planes about the line $\overleftrightarrow{DE}$, and hence, $M_2 \neq M_3$ by Lemma 2.1(h). This implies that all four points M_1, M_2, M_3 and M are distinct and lie on the line l^*. Furthermore, since M_3 and the set of points $\{M_1, M, M_2\}$ lie in different half-planes about the line $\overleftrightarrow{DE}$, the order of the points on the line l^* has to be as indicated in Figure 2.6:

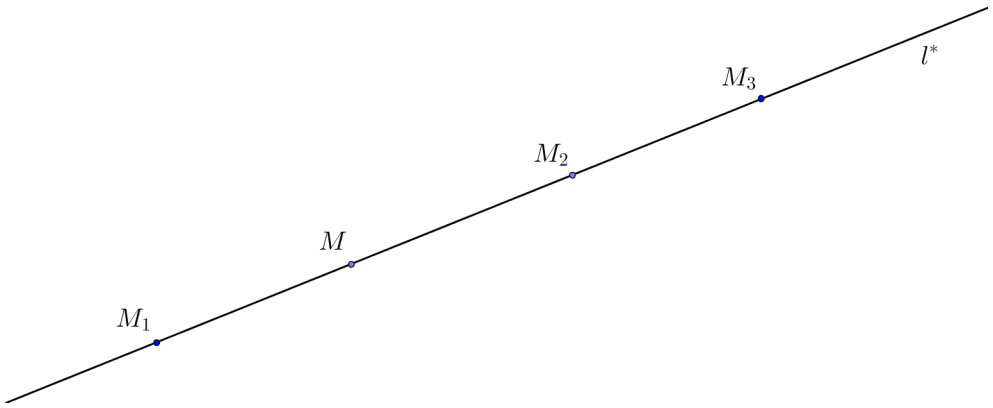


Figure 2.6: Order of points M_1, M_2, M_3 and M

The following construction leads us to a contradiction to the order of points M, M_1, M_2, M_3 shown in Figure 2.6.

Depending on the type of convexity of γ (*i.e.,* whether $\Phi_{\angle A}(\gamma)$ or $\overline{S(\angle A)\backslash\Phi_{\angle A}(\gamma)}$ is a convex set), we now define the point $Z \in AE$ such that XZ separates Φ_2 and Φ_3:

- $\Phi_{\angle A}(\gamma)$ *is convex.* By Z we denote the point of intersection of γ and AE.

- $\overline{S(\angle A)\backslash\Phi_{\angle A}(\gamma)}$ *is convex.* First, by T we denote the point of intersection of γ with the side of $\angle A$ containing point D. The point Z is then defined as the point of intersection of the lines $\overleftrightarrow{XT}$ and $\overleftrightarrow{AE}$.

Let the points $X' \in EX$ and $Z' \in EZ$ be such that $|EX'| = 2|X'X|$ and $|EZ'| = 2|ZZ'|$. Also, denote by D' and A' the points on DE and AE, respectively, such that $|ED'| = 2|DD'|$ and $|EA'| = 2|AA'|$. The inequalities $|EX| < |ED|$ and $|EZ| \leq |EA|$ imply that $|EX'| < |ED'|$ and $|EZ'| \leq |EA'|$, and thus, $D'A'$ and the point E lie in different half-planes about the line $\overleftrightarrow{X'Z'}$. Lemma 2.3 implies that $M_2 \in \triangle EX'Z'$ while by Lemma 2.1(c), $M \in D'A'$. An immediate conclusion is that points M and M_2 lie in different half-planes about the line $\overleftrightarrow{X'Z'}$. In addition, due to the convexity of γ, M_3 and M_2 lie in different half-planes about $\overleftrightarrow{XZ}$ and hence, about $\overleftrightarrow{X'Z'}$ as well. We can now draw the conclusion that M_2 and the set of points $\{M, M_3\}$ lie in different half-planes about the line $\overleftrightarrow{X'Z'}$, which contradicts the order of points in Figure 2.6 (indeed, according to the order of points shown on Figure 2.6, $M_2 \in MM_3$, and thus, if both points M and M_3 lie in the same half-plane, M_2 must lie in that same half-plane as well). This contradiction implies that our assumption about the possibility of cases $(1')$ and $(2')$ is false, and thus, case $(3')$ must take place, which is exactly what was required to be proven in the claim of the lemma. □

The results of Lemmas 2.6 and 2.7 can be combined as follows:

Corollary 2.1. *Let $\gamma \in \Gamma(\angle A)$ be a convex curve inside some angle $\angle A$, and let $DE := DE(\angle A, \gamma)$ be the straight line segment approximation of the curve γ defined by* (2.6)*. Then*

$$\#(\gamma \cap DE) = \begin{cases} \infty, \text{ if } \gamma \text{ is a segment of a straight line,} \\ \infty, \text{ if } \gamma = \partial(\triangle AXY),\ \triangle AXY \subset S(\angle A) \text{ such that } |XK| = |YL|,\ \{K, L\} = \overleftrightarrow{XY} \cap \angle A, \\ 2, \text{ otherwise.} \end{cases}$$

Remark 2.9. *In the proof of Lemma 2.7 and in Corollary 2.1, equality $\#(\gamma \cap DE) = \infty$ is interpreted as the set of points of intersection of γ and DE is infinite, even though the cardinality of the set $\gamma \cap DE$, in that case, is the continuum.*

Corollary 2.1 states that $\#(\gamma \cap DE) = 2$ except for some triangular or linear curves γ. For simplicity of further analysis, for a fixed angle $\angle A$, we introduce the set

$$\Gamma_0(A) := \{\gamma \in \Gamma(\angle A) : \gamma \text{ is convex}, \#(\gamma \cap DE) = 2\}.$$

One can establish further results regarding the way DE cuts the set $\Phi_{\angle A}(\gamma)$. It turns out that this cut is, in some sense, balanced:

Lemma 2.8. *Given $\angle A$ on the plane and a curve $\gamma \in \Gamma(\angle A)$ such that $\gamma \subset \partial\Omega$ for some convex set Ω, the midpoint of the segment $DE = DE(\angle A, \gamma)$ defined via Construction 2.2 always lies in Ω.*

Proof. Let P and Q be the points of intersection of γ and $\angle A$. By Lemma 2.3, the centroid of $\triangle ADE$ is in the set $\mathcal{R}$ defined via (2.3). This immediately implies that the midpoint of DE is in $Seg(\gamma, P, Q)$. Since Ω is convex, $Seg(\gamma, P, Q) \subset \Omega$ and the claim of Lemma 2.8 follows. □

Now that we have established the geometric interrelation between DE and γ, we present some quantitative results of this approximation. Since we expect DE to provide a sufficiently good approximation of γ, we want to control the areas of $\overline{\Phi_{\angle A}(\gamma) \backslash \triangle ADE}$ and $\overline{\triangle ADE \backslash \Phi_{\angle A}(\gamma)}$. It turns out that, depending on which of the sets $\Phi_{\angle A}(\gamma)$ or $\overline{S(\angle A) \backslash \Phi_{\angle A}(\gamma)}$ is convex, we can provide various estimates of these areas.

Notation 2.2. *Given a bounded region Φ in $\mathbb{R}^2$, $\mathcal{A}(\Phi)$ denotes the area of this region. Given a simple convex curve γ and two points $P, Q \in \gamma$ such that $PQ \cap \gamma = \{P, Q\}$,*

$$\mathcal{A}(\gamma, P, Q) = \mathcal{A}\left(Seg\left(\gamma, P, Q\right)\right).$$

Lemma 2.9 (Positive result)**.** *Suppose that $\gamma \in \Gamma_0(\angle A)$ for some $\angle A$ in the plane and $\Phi_{\angle A}(\gamma)$ is convex. Let D' and E' be the points of intersection of DE and γ, where $DE = DE(\angle A, \gamma)$ is the straight line segment approximation defined in Construction 2.2. Then,*

$$\mathcal{A}(\gamma, D', E') \leq \frac{5}{9}\mathcal{A}(\Phi_{\angle A}(\gamma)).$$

Proof. Let M be the centroid of both $\triangle ADE$ and $\Phi_{\angle A}(\gamma)$ (by construction, the centroids of both sets coincide). Draw the line l parallel to DE and passing through M. l divides $\Phi_{\angle A}(\gamma)$ into two sets Φ_1 and Φ_2, and assume that $A \in \Phi_1$. Then since $Seg(\gamma, D', E') \subset \Phi_2$,

$$\mathcal{A}(\gamma, D', E') \leq \mathcal{A}(\Phi_2). \tag{2.8}$$

Lemma 2.2 (Winternitz theorem) implies

$$\frac{\mathcal{A}(\Phi_1)}{\mathcal{A}(\Phi_2)} \geq \frac{4}{5},$$

and hence,

$$\frac{\mathcal{A}(\Phi_2)}{\mathcal{A}(\Phi_A(\gamma))} = \frac{\mathcal{A}(\Phi_2)}{\mathcal{A}(\Phi_1) + \mathcal{A}(\Phi_2)} = \frac{1}{\frac{\mathcal{A}(\Phi_1)}{\mathcal{A}(\Phi_2)} + 1} \leq \frac{1}{4/5 + 1} = \frac{5}{9}. \tag{2.9}$$

Combining equations (2.8) and (2.9), we obtain the desired inequality. □

The next example shows that there is no similar result for the curves γ such that $\overline{S(\angle A) \backslash \Phi_{\angle A}(\gamma)}$ is convex.

Remark 2.10 (Negative example)**.** *For any given $\angle A$,*

$$\sup_{\{\gamma \in \Gamma_0(\angle A):\, \overline{S(\angle A)\backslash \Phi_{\angle A}(\gamma)} - convex\}} \frac{\mathcal{A}(\gamma, D', E')}{\mathcal{A}\left(\Phi_{\angle A}(\gamma)\right)} = \infty, \tag{2.10}$$

where D' and E' are the points of intersection of $\gamma \in \Gamma_0(\angle A)$ and its straight line segment approximation $DE = DE(\angle A, \gamma)$ defined in Construction 2.2.

Proof. Let K, L be two points on the different sides of $\angle A$ such that $|AK| = |AL|$, and let $\angle KAL =: \beta$ and F be the midpoint of KL. For every h, $0 < h \leq |AF|$, we construct $\gamma_h \in \Gamma_0(\angle A)$ and its straight line segment approximation $D_h E_h$ with $D_h \in AK$ and $E_h \in AL$ (we use the subscript h to emphasize the dependence on the parameter h).

Choose O_h to be the point on the median AF of $\triangle AKL$ such that $|AO_h| := h$. Now, let $\gamma_h := KO_h \cup LO_h$. It is evident that the set $\overline{S(\angle A) \backslash \Phi_{\angle A}\left(\gamma_h\right)}$ is convex, for all $0 < h \leq |AF|$. We now show that

$$\lim_{h \to 0^+} \frac{\mathcal{A}(\gamma_h, D_h', E_h')}{\mathcal{A}\left(\Phi_{\angle A}\left(\gamma_h\right)\right)} = \infty. \tag{2.11}$$

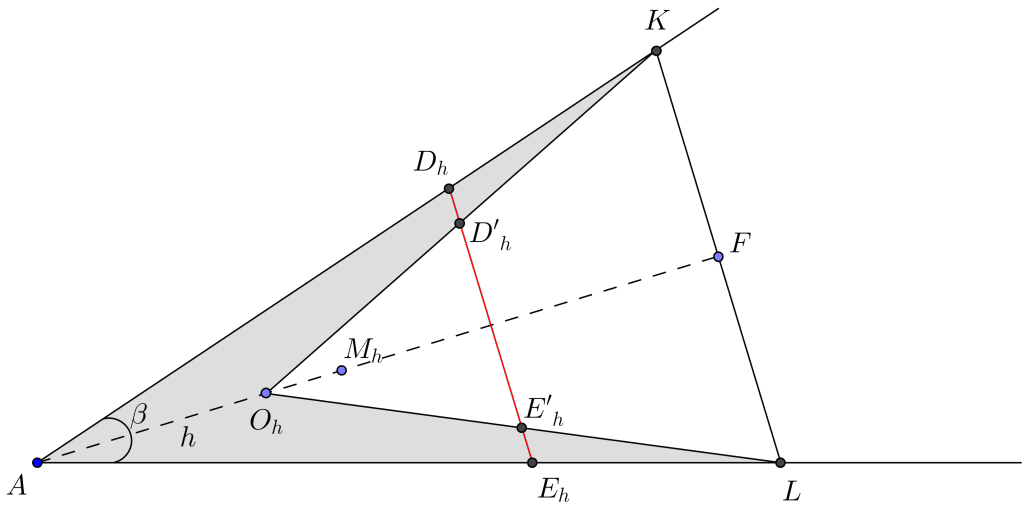


Figure 2.7: $\overline{S(\angle A)\backslash\Phi_{\angle A}(\gamma)}$ is convex: the ratio $\frac{\mathcal{A}(\overline{\triangle AD_hE_h\backslash\Phi_{\angle A}(\gamma_h)})}{\mathcal{A}(\Phi_{\angle A}(\gamma_h))}$ can be arbitrarily large

Indeed, it is easy to see that

$$\mathcal{A}\left(\Phi_{\angle A}\left(\gamma_h\right)\right) = 2\left(\frac{1}{2}|AK|h\sin\frac{\beta}{2}\right) = |AK|h\sin\frac{\beta}{2},$$

and, hence,

$$\lim_{h\to 0^+}\mathcal{A}\left(\Phi_{\angle A}\left(\gamma_h\right)\right) = 0. \tag{2.12}$$

Let M_h denote the centroid of $\Phi_{\angle A}\left(\gamma_h\right)$. Since M_h is the arithmetic mean of the centroids of $\triangle AO_hK$ and $\triangle AO_hL$, one can show that $M_h \in AF$ and $|AM_h| = (|AF|+h)/3$. From this, we can conclude that $\lim\limits_{h\to 0^+}|O_hD'_h| = \lim\limits_{h\to 0^+}|O_hE'_h| = |AK|/2$, and therefore,

$$\lim_{h\to 0^+}\mathcal{A}(\gamma_h, D'_h, E'_h) = \frac{1}{2}\left(\frac{|AK|}{2}\right)^2\sin\beta = \frac{1}{4}\mathcal{A}(\triangle AKL). \tag{2.13}$$

The continuity of $\mathcal{A}\left(\Phi_{\angle A}\left(\gamma_h\right)\right)$ and $\mathcal{A}(\gamma_h, D'_h, E'_h)$, as functions of h, together with (2.12) and (2.13), imply (2.11), and (2.10) follows. □

However, one can establish another kind of estimate on the proposed approximation of γ of the type considered in Remark 2.10.

Lemma 2.10 (Positive result)**.** *Let $\angle A$ be given and assume that $\gamma \in \Gamma_0(\angle A)$ intersects $\angle A$ at the points P and Q. In addition, assume that the set $\overline{S(\angle A)\backslash\Phi_{\angle A}(\gamma)}$ is convex and $DE = DE(\angle A, \gamma)$ is the straight line segment approximation of γ defined in Construction 2.2. Then*

$$\mathcal{A}(\overline{\Phi_{\angle A}(\gamma)\backslash\triangle ADE}) \leq \mathcal{A}(\overline{\triangle ADE\backslash\Phi_{\angle A}(\gamma)}) \leq \mathcal{A}(\gamma, P, Q).$$

Proof. The second inequality trivially follows from the convexity of γ and the fact that $\gamma(D', E') \subset \gamma$, where D' and E' are the points of intersection of DE and γ.

We now prove the first inequality. Let M_1, M_2 and M_3 be the centroids of $\Phi_1 := \Phi_{\angle A}(\gamma) \cap \triangle ADE$, $\Phi_2 := \overline{\triangle ADE\backslash\Phi_{\angle A}(\gamma)}$ and $\Phi_3 := \overline{\Phi_{\angle A}(\gamma)\backslash\triangle ADE}$, respectively. Also, by M we denote the centroid of $\triangle ADE$. Since the centroids of $\triangle ADE = \Phi_1 \cup \Phi_2$ and $\Phi_{\angle A}(\gamma) = \Phi_1 \cup \Phi_3$ coincide, and no two sets Φ_i and Φ_j $(i \neq j)$ have a common interior point, from Lemma 2.1(i) we deduce that M_1, M_2, M_3 and M lie on the same line and M lies between M_1 and M_2 as well as between M_1 and M_3. Furthermore, since the regions Φ_1 and Φ_2 are separated from Φ_3 by the segment DE, M_2 lies between M_1 and M_3 (we actually have the same order of these points as shown in Figure 2.6). Applying the Decomposition Rule (Lemma 2.1(b)), we obtain

$$|M_1M|\,\mathcal{A}(\Phi_1) = |M_2M|\,\mathcal{A}(\Phi_2) = |M_3M|\,\mathcal{A}(\Phi_3). \tag{2.14}$$

Taking into account that $|M_2M| < |M_3M|$, we conclude that

$$\mathcal{A}(\Phi_3) < \mathcal{A}(\Phi_2), \tag{2.15}$$

and the first inequality follows. □

Although Lemma 2.9 already provides some quantitative estimates on the approximation of γ by the straight line segment DE defined in Construction 2.2 in the case where $\Phi_{\angle A}(\gamma)$ is convex, the next example shows that one cannot establish a result similar to Lemma 2.10 in that case. More precisely, the set $\overline{\triangle ADE\backslash\Phi_{\angle A}(\gamma)}$ can have arbitrarily large area while the area of the set $\overline{\triangle\Phi_{\angle A}(\gamma)\backslash\triangle ADE}$ can be arbitrarily small.

Remark 2.11 (Negative example)**.** *For any given $\angle A$,*

$$\sup_{\{\gamma\in\Gamma_0(\angle A):\,\Phi_{\angle A}(\gamma)-convex\}} \frac{\mathcal{A}(\overline{\triangle ADE\backslash\Phi_{\angle A}(\gamma)})}{\mathcal{A}(\gamma, D', E')} = \infty, \tag{2.16}$$

where P, Q are the points of intersection of $\gamma \in \Gamma_0(\angle A)$ and the sides of angle $\angle A$, $DE = DE(\angle A, \gamma)$ is the straight line segment approximation of γ defined in Construction 2.2, and D', E' are the points of intersection of DE and γ.

Proof. Given angle $\angle A$, choose the points K and L on the sides of the angle such that $|AK| = |AL| > 0$, and let N be the midpoint of KL. For every h, $0 < h \leq |AK|$, consider the following construction for γ_h (once again, the subscript h denotes the dependence on the parameter h):

1. Let $P_h \in AK$ and $Q_h \in AL$ be such that $|AP_h| = |AQ_h| := h$.
2. $\gamma_h := P_hN \cup Q_hN$.

Then, for $0 < h \leq |AK|$, the set $\Phi_{\angle A}(\gamma_h)$ is convex. We will show that

$$\lim_{h\to 0^+} \frac{\mathcal{A}(\overline{\triangle AD_hE_h\backslash\Phi_{\angle A}(\gamma_h)})}{\mathcal{A}(\triangle P_hQ_hN)} = \infty. \tag{2.17}$$

Indeed, it is easy to see that

$$\mathcal{A}(\triangle P_hQ_hN) < \frac{1}{2}|P_hQ_h||AN| < \frac{1}{2}(|AP_h| + |AQ_h|)|AN| = |AN|h,$$

and therefore,

$$\lim_{h\to 0^+} \mathcal{A}(\triangle P_hQ_hN) = 0. \tag{2.18}$$

Let M_h, M_h' and M_h'' be the centroids of the quadrilateral $AP_hNQ_h, \triangle AP_hQ_h$ and $\triangle P_hQ_hN$. Note that since $\triangle AKL$ and $\triangle AP_hQ_h$ are isosceles, the points M_h, M_h' and M_h'' lie on the median AN of $\triangle AKL$. Let h_1 be the length of the median of $\triangle AP_hQ_h$ towards the side P_hQ_h. The Decomposition Rule (Lemma 2.1(b)) implies that

$$|M_hM_h'|\mathcal{A}(\triangle AP_hQ_h) = |M_hM_h''|\mathcal{A}(\triangle P_hQ_hN) \Leftrightarrow |M_hM_h'|h_1 = |M_hM_h''|(|AN| - h_1),$$

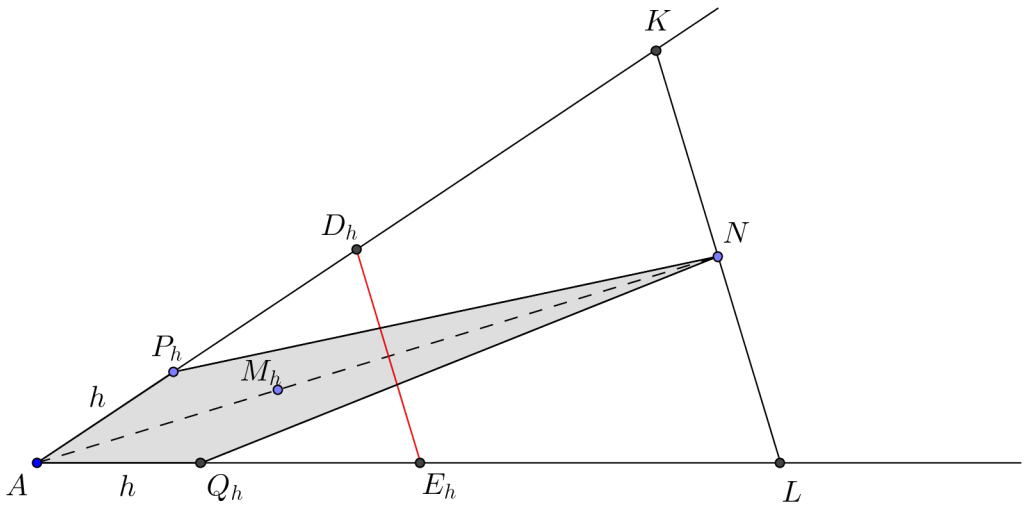


Figure 2.8: $\Phi_{\angle A}(\gamma)$ is convex: the ratio $\frac{\mathcal{A}(\overline{\triangle AD_hE_h \setminus \Phi_{\angle A}(\gamma_h)})}{\mathcal{A}(\triangle P_hQ_hN)}$ can be arbitrarily large

and since clearly $\lim_{h\to 0^+} h_1 = 0$, we obtain that

$$\lim_{h\to 0^+} |M_hM_h''| = 0,$$

and hence,

$$\lim_{h\to 0^+} |AM_h| = \lim_{h\to 0^+} |AM_h''| = \lim_{h\to 0^+} [h_1 + (|AN| - h_1)/3] = \frac{|AN|}{3}.$$

We can conclude from the last equality that $\lim_{h\to 0^+} |D_hK'| = \lim_{h\to 0^+} |E_hL'| = 0$, where K' and L' are the midpoints of AK and AL, respectively. Then,

$$\lim_{h\to 0^+} \mathcal{A}(\overline{\triangle AD_hE_h \setminus \Phi_{\angle A}(\gamma_h)}) = \lim_{h\to 0^+} \mathcal{A}(\triangle AD_hE_h) = \mathcal{A}(\triangle AK'L') = \frac{1}{4}\mathcal{A}(\triangle AKL) > 0, \tag{2.19}$$

whereas

$$0 \le \lim_{h\to 0^+} \mathcal{A}(\triangle AD_hE_h \cap \Phi_{\angle A}(\gamma_h)) \le \lim_{h\to 0^+} \big[\mathcal{A}(\triangle AP_hQ_h) + \mathcal{A}(\triangle P_hQ_hN)\big] = 0.$$

The continuity of $\mathcal{A}(\triangle P_hQ_hN)$ and $\mathcal{A}(\overline{\triangle AD_hE_h \setminus \Phi_{\angle A}(\gamma_h)})$, as functions of h, together with (2.18) and (2.19), imply (2.17), and (2.16) follows since clearly $\mathcal{A}(\gamma_h, D_h', E_h') \le \mathcal{A}(\triangle P_hQ_hN)$. □

Remark 2.12. *Following the notations introduced in the proof of Remark 2.11, we note that if we can further split $\triangle AD_hE_h$ into two triangles $\triangle AD_hF$ and $\triangle AE_hF$ such that $(NP_h \cap \triangle AD_hE_h) \subset \triangle AD_hF$ and $(NQ_h \cap \triangle AD_hE_h) \subset \triangle AE_hF$, the next application of our approximation of the curves inside the $\angle AD_hF$ and $\angle AE_hF$ will fully reconstruct each of the two segments of γ_h inside $\triangle AD_hE_h$.*

These properties conclude the analysis of the proposed approximation method, and thus we move to the final algorithm.

2.3 General Algorithm and Its Properties

In this section, we introduce the subdivision algorithm for our adaptive triangulation method to approximate the characteristic function

$$f(x) := \chi_\Omega(x), \quad x \in [0,1]^2,$$

where Ω is a closed convex set contained in $[0,1]^2$ with a piecewise-smooth boundary.

A general approach in constructing adaptive triangulations is described on pages 30-31, under the paragraph "General Algorithm". According to that algorithm, before implementing any subdivision, we need to determine which triangles in a triangulation we subdivide and which we do not, *i.e.*, we need to define the *decision function* $e(\Delta)$. In our setting, a suitable choice for $e(\Delta)$ is $E_1(f,\Delta)_1$, the error of best L_1-approximation of f by constants over Δ. In other words, if for simplicity of notation, we set

$$\varepsilon(S) := E_1(f,S)_1 = \min\{\mathcal{A}(S \cap \Omega), \mathcal{A}(S \backslash \Omega)\}, \quad S \subset \mathbb{R}^2, \tag{2.20}$$

we choose $e(\Delta)$ to be $\varepsilon(\Delta)$.

Now given a precision $\epsilon > 0$ and an initial triangulation $\mathcal{D}_0$, we construct a sequence of hierarchial triangulations as follows: for every $k \geq 0$, we obtain the triangulation $\mathcal{D}_{k+1}$ from $\mathcal{D}_k$ by subdividing the triangles $\Delta \in \mathcal{D}_k$ with $\varepsilon(\Delta) > \epsilon$ *(we do not necessarily subdivide all $\Delta \in \mathcal{D}_k$)*. In the following Subsection 2.3.1, we define the subdivision rule for the triangles Δ that require a further split (*i.e.*, with $\varepsilon(\Delta) > \epsilon$).

2.3.1 Main Subdivision Rule

Given a triangle Δ such that $\varepsilon(\Delta) > \epsilon$, we decide on the type of subdivision based on the values of the function at the vertices v_1, v_2, v_3. Thus, we define the following types of triangles

$$Tr_i(\Omega) := \{\Delta := \triangle v_1 v_2 v_3 : \sum_{k=1}^{3} f(v_k) = i\}, \qquad i = 0, 1, 2, 3.$$

Remark 2.13. *Note that for every $\Delta \in Tr_3(\Omega)$, $v_k \in \Omega\,(k = 1, 2, 3)$, and hence, since Ω is convex, $\Delta \subset \Omega$ and $\varepsilon(\Delta) = 0$. This implies that if $\varepsilon(\Delta) > \epsilon > 0$, $\Delta \notin Tr_3(\Omega)$.*

We also distinguish a particular subset of $Tr_1(\Omega)$:

$$Tr_1^*(\Omega) := \{\Delta \in Tr_1(\Omega) : f(v_j) = 1,\ \#(\Omega \cap \overline{v_k v_l}) \leq 1, k, l \neq j\}.$$

In other words, for triangles $\Delta \in Tr_1(\Omega)$, we distinguish two cases: when $\Delta \cap \partial\Omega$ is a connected curve, and when it is not. It is our idea to approximate $\Delta \cap \partial\Omega$ by the straight line segment approximation DE defined in (2.6). If $\Delta \cap \partial\Omega$ is not a connected curve (and $f(v_j) = 1$), the straight line segment DE approximates

$$(\Delta \cap \partial\Omega) \cup (\Omega \cap \overline{v_k v_l}), \quad k, l \neq j,$$

and the second set in the union is not a part of $\partial\Omega$. That is why in our subdivision algorithm, if $\Delta \cap \partial\Omega$ is not a connected curve, we first split Δ into two triangles Δ_1, Δ_2 such that the curves $\Delta_i \cap \partial\Omega$, $i = 1, 2$, are both connected, and then construct the straight line segment approximation for each triangle separately.

Remark 2.14. *Note that for any $\Delta \in Tr_1^*(\Omega) \cup Tr_2(\Omega)$, $\Delta \cap \partial\Omega$ is a connected curve due to the convexity of Ω.*

Now we are ready to precisely describe our subdivision algorithm.

Main Subdivision Algorithm:

Preliminary Step: Given Δ as the input triangle, we first verify whether $\varepsilon(\Delta) > \epsilon$. If $\varepsilon(\Delta) \leq \epsilon$, no subdivision is required; otherwise, we determine $i \in \{0, 1, 2\}$ such that $\Delta \in Tr_i(\Omega)$. The subdivision rule is now described for every set of triangles $Tr_i(\Omega)$, $i = 0, 1, 2$ ($i \neq 3$ due to Remark 2.13).

$\boxed{\Delta \in Tr_0(\Omega)}$. Label the vertices with the labels A, B, C randomly in the clockwise order. We proceed as follows *(note that $\Delta \cap \Omega \neq \emptyset$, since otherwise $\varepsilon(\Delta)$ would have to be 0)*:

We find the centroid of the closed region $\Delta \cap \Omega$, point M, and split Δ into 3 triangles: $\triangle AMB, \triangle AMC$ and $\triangle BMC$ (see Figure 2.9). Note that since $\Delta \cap \Omega$ is convex, $f(M) = 1$, and the sum of the values of the function at the vertices of each child is equal to 1.

Children: $\mathcal{C}(\Delta) \subset Tr_1(\Omega)$.

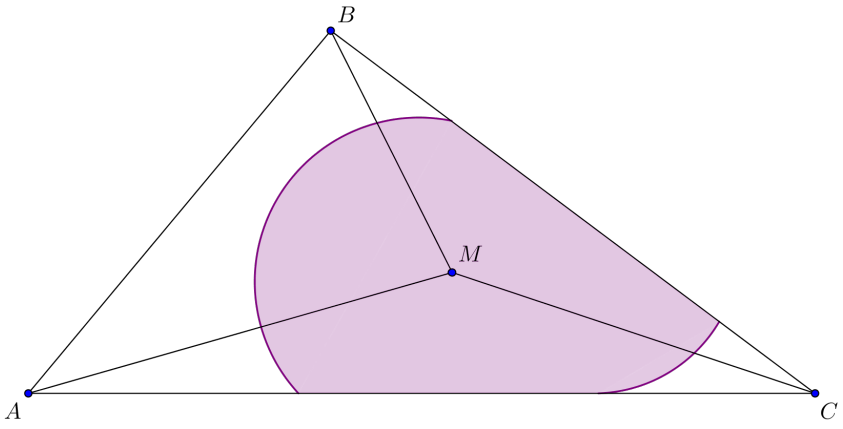


Figure 2.9: Subdivision of $\Delta \in Tr_0(\Omega)$

$\boxed{\Delta \in Tr_1(\Omega)}$. Label the vertices with the labels A, B, C in the clockwise order so that $f(A) = 1$. We now distinguish two cases: $\Delta \in Tr_1^*(\Omega)$ and $\Delta \notin Tr_1^*(\Omega)$. *(To determine which of the cases takes place, for instance, we may calculate the line integral of f over the straight line segment BC, or the 2-dimensional integral of f over $[0, 1]^2 \cap \Phi$, where Φ is the semiplane of points about the line BC that does not contain the point A.)*

- $\Delta \in Tr_1^*(\Omega)$. Then, as mentioned in Remark 2.14, $\gamma := \Delta \cap \partial\Omega$ is a connected curve, and since Ω is convex, $\gamma \in \Gamma(\angle BAC)$. We construct the straight line segment approximation DE of γ such that $D \in \overrightarrow{(A,B)}$ and $E \in \overrightarrow{(A,C)}$. Note that one of these two points may lie outside Δ.

  1. Denote by F the midpoint of DE. *Note that $F \in \Omega$ by Lemma 2.8 (as shown in Figures 2.10 and 2.11).*

  2. Let $\gamma_1 := \triangle ADF \cap \partial\Omega$ and $\gamma_2 := \triangle AEF \cap \partial\Omega$. Note that $\gamma_1 \in \Gamma(\angle ADF)$ and $\gamma_2 \in \Gamma(\angle AEF)$. We now construct straight line segment approximations D_1D_2 and E_1E_2 of γ_1 and γ_2, respectively ($D_1 \in \overrightarrow{(A,B)}$ and $E_1 \in \overrightarrow{(A,C)}$; note that we use distinct, different from 'common DE', labels for the straight line segment approximations constructed for different parts of $\Delta \cap \partial\Omega$). Once again, one of the points D_1, D_2, E_1, E_2 may lie outside Δ.

  The segments DE, AF, D_1D_2 and E_1E_2 subdivide Δ into five (polygonal but not necessarily triangular) regions. We denote them

$$\Phi_1 := \overline{\Delta \backslash \triangle ADE},\ \Phi_2 := \triangle DD_1D_2,\ \Phi_3 := \triangle EE_1E_2, \Phi_4 := AFD_2D_1,\ \Phi_5 := AFE_2E_1.$$

  Note that Δ can be a proper subset of $\bigcup_{i=1}^{5} \Phi_i$, because $\triangle ADE$ may not entirely lie in Δ. However, we make our error analysis based on the local error of approximation of f over these regions.

  3. Let

$$\delta_1 := \max\{\mathcal{A}(\Phi_1 \cap \Omega), \mathcal{A}(\Phi_2 \cap \Omega), \mathcal{A}(\Phi_3 \cap \Omega)\}.$$

  In order to guarantee that the error of approximation of f over the children Δ' of Δ is decreasing (at a certain rate) compared to the error of approximation of f over Δ, we want to bound the quantity $\varepsilon(\Delta')/\varepsilon(\Delta)$ (see more details in Remark 2.16). In particular, we aim to have bounds for

$$\frac{\delta_1}{\mathcal{A}(\overline{\Delta \backslash \Omega})} \text{ and } \frac{\delta_1}{\mathcal{A}(\Delta \cap \Omega)}$$

  from above. The second ratio can be bounded by $5/9$ from above using Lemma 2.2 (Winternitz Theorem) (see the details in Subsection 2.3.2). Hence, we consider two further subcases when $\frac{\delta_1}{\mathcal{A}(\overline{\Delta \backslash \Omega})} \leq 5/9$, and when $\frac{\delta_1}{\mathcal{A}(\overline{\Delta \backslash \Omega})} > 5/9$.

Case (1) If $\delta_1 \leq \frac{5}{9}\mathcal{A}(\overline{\Delta \backslash \Omega})$, then we perform the following steps to 'complete' our subdivision, *i.e.*, split Δ into triangles (from $Tr_1^*(\Omega)$ and $Tr_2(\Omega)$):

3.1. Denote by M_1, M_2 and M_3 the centroids of $\Phi_1 \cap \Omega$, $\Phi_2 \cap \Omega$, $\Phi_3 \cap \Omega$, respectively. $M_i \in \Delta$, $i = 1, 2, 3$.

3.2. Let F_1 and F_2 be the midpoints of D_1D_2 and E_1E_2, respectively.

3.3. Denote by $\widetilde{\Phi}_i := \Phi_i \cap \Delta$, $i = 2, 3, 4, 5$ (maximum two of Φ_i's are not entirely in Δ). Note that each $\widetilde{\Phi}_i$ is a polygon.

3.4. Split polygon Φ_1 into triangles by joining M_1 with all of its vertices and F. Split polygon $\widetilde{\Phi_2}$ by joining M_2 with all of its vertices and F_1. Split polygon $\widetilde{\Phi_3}$ by joining M_3 with all of its vertices and F_2. Split polygons $\widetilde{\Phi_4}$ and $\widetilde{\Phi_5}$ by joining A and F with F_1 and F_2.

If D and E both lie inside Δ, Step 3.4 provides the final split into 19 triangles (see Figure 2.10): $\triangle BM_1C, \triangle BM_1D, \triangle EM_1C, \triangle FM_1D,$ $\triangle FM_1E, \triangle M_2D_1F_1, \triangle M_2F_1D_2, \triangle DM_2D_2, \triangle DD_1M_2, \triangle M_3E_1F_2, \triangle M_3F_2E_2,$ $\triangle EM_3E_2, \triangle EE_1M_3, \triangle AF_1D_1, \triangle AF_1F, \triangle AFF_2, \triangle AF_2E_1, \triangle FE_2F_2, \triangle F_1D_2F.$

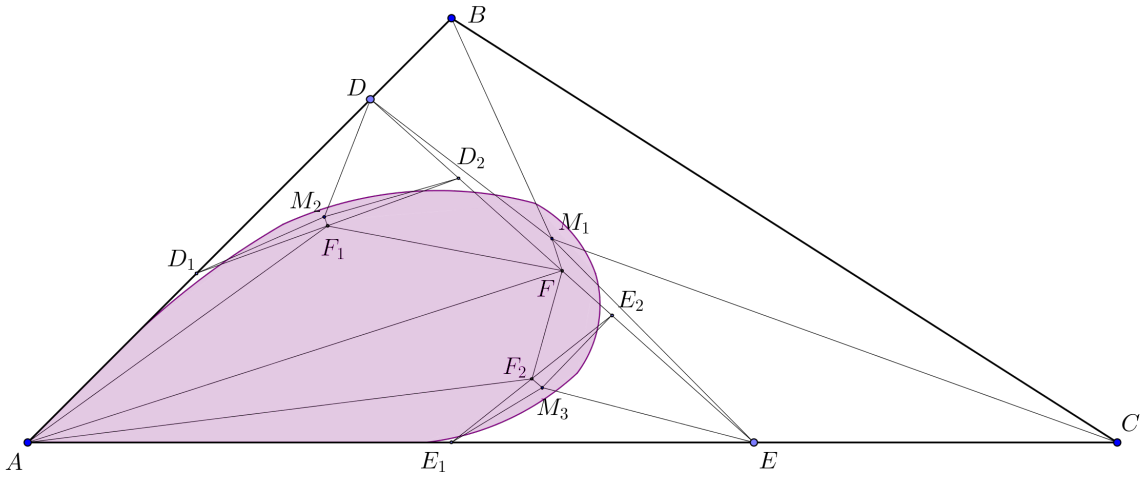


Figure 2.10: Subdivision of $\Delta \in Tr_1^*(\Omega)$, case (1)

If one of the points D or E lies outside Δ, but D_1D_2 and E_1E_2 are entirely in Δ, $\widetilde{\Phi_4}$, $\widetilde{\Phi_5}$ are quadrilaterals, and Step 3.4 provides the final split into 19 triangles as well.

If one of the points D_1, D_2, E_1, E_2 lies outside Δ, one of $\widetilde{\Phi_4}$, $\widetilde{\Phi_5}$ is now a pentagon. Without loss of generality, assume that either E_1 or E_2 is outside Δ. We complete Step 3.4 by joining F_2 with the vertex of the pentagon $\widetilde{\Phi_5}$ on the side BC. The final split consists of 19 triangles as well (see Figure 2.11). *Children:* $\mathcal{C}(\Delta) \subset \big(Tr_1^*(\Omega) \cup Tr_2(\Omega) \cup Tr_3(\Omega)\big)$.

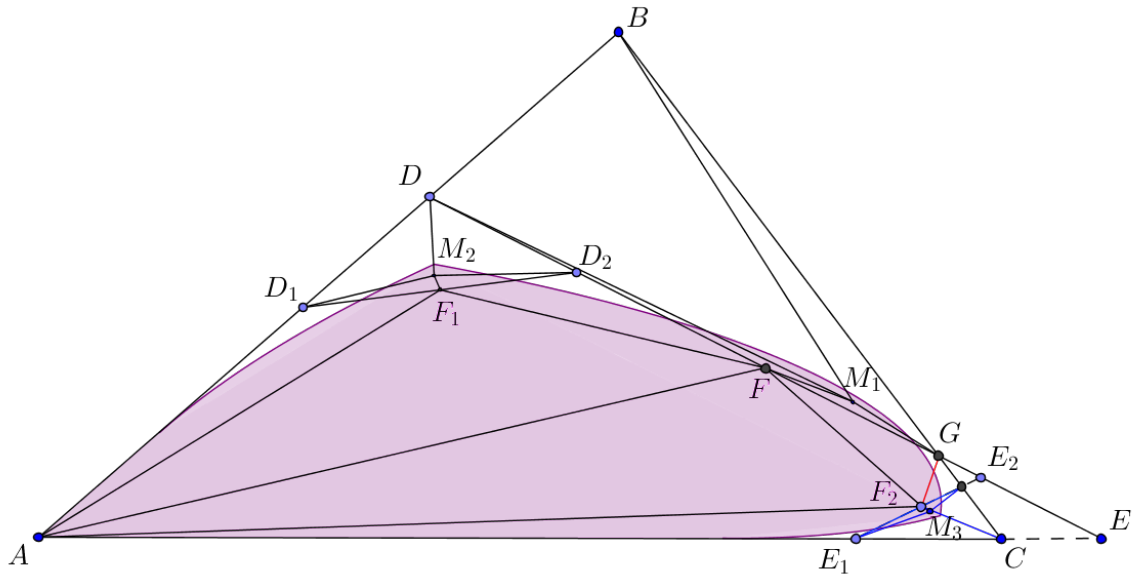


Figure 2.11: Subdivision of $\Delta \in Tr_1^*(\Omega)$, case (1), points E and E_2 lie outside Δ

Case (2) If $\delta_1 > \frac{5}{9}\mathcal{A}(\overline{\Delta \backslash \Omega})$, then disregard Steps 1,2,3,3.1–3.4, and 3.5. Find a point $H \in BC$ such that

$$\max\{\mathcal{A}(\overline{\triangle AHB \backslash \Omega}), \mathcal{A}(\overline{\triangle AHC \backslash \Omega})\} \leq \frac{5}{9}\mathcal{A}(\overline{\Delta \backslash \Omega}).$$

The algorithm for finding such a point can be as follows:

Construction 2.3. *Step 1. Choose a natural number n such that*

$$\mathcal{A}(\overline{\Delta \backslash \Omega}) \geq \frac{9\mathcal{A}(\Delta)}{n},$$

and denote by $H_1, \ldots, H_{n-1}$ the points that divide BC into n equal parts, i.e.,

$$|H_{i-1}H_i| = \frac{|BC|}{n}, \quad 1 \leq i \leq n, \quad (H_0 := B, H_n := C).$$

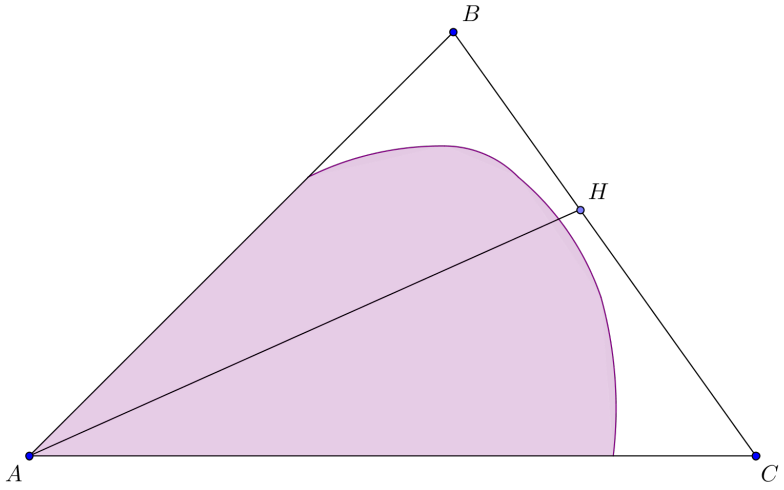


Figure 2.12: Subdivision of $\Delta \in Tr_1^*(\Omega)$, case (2)

Step 2. Find $i^* := \min\{i : \mathcal{A}(\overline{\triangle AH_iB \backslash \Omega}) \geq \frac{4}{9}\mathcal{A}(\overline{\Delta \backslash \Omega})\}$.

The point $H := H_{i^*}$ *will have all the necessary properties.*

3.6. The final split is now $\triangle AHB$ and $\triangle AHC$ (see Figure 2.12).

Children: $\mathcal{C}(\Delta) \subset Tr_1^*(\Omega)$.

- $\Delta \notin Tr_1^*(\Omega)$, or in other words, $\Delta \in Tr_1(\Omega), f(A) = 1$ and $BC \cap \Omega$ contains more than one point. Then, we find a point $G \in BC$ such that $f(G) = 1$. This can be done, for instance, by:

  1. Finding the centroids M, M_0 of $\Delta \cap \Omega$ and the part of Ω lying in the half-plane about the line $\overleftrightarrow{BC}$ that does not contain point A.

  2. Finding the point of intersection of MM_0 and BC. Δ is to be split into $\triangle ABG$ and $\triangle ACG$ (see Figure 2.13). *If* Ω *is contained in the same half-plane about* $\overleftrightarrow{BC}$ *as the point* A, G *can be found as the centroid of the* 1*-dimensional set* $BC \cap \Omega$.

  Children: $\mathcal{C}(\Delta) \subset Tr_2(\Omega)$.

$\boxed{\Delta \in Tr_2(\Omega)}$. Label the vertices of Δ with the labels A, B, C in the clockwise order so that $f(A) = 0$. The next procedure is as follows (refer to Figures 2.14 and 2.15):

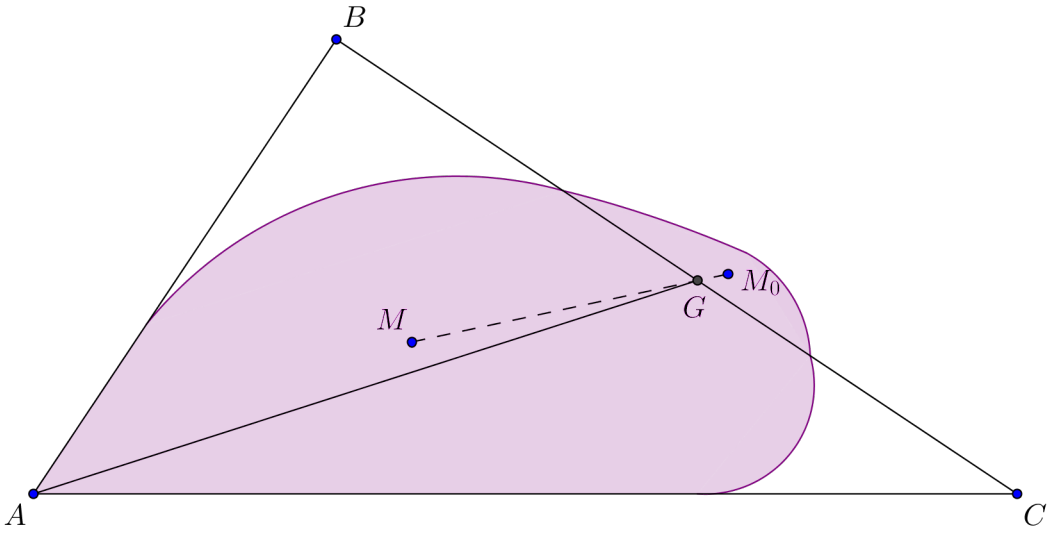


Figure 2.13: Subdivision of $\Delta \in Tr_1(\Omega)\backslash Tr_1^*(\Omega)$

1. Denote by M the centroid of $\Delta \cap \Omega$. Let N be the point of intersection of AM and BC.

2. Note that $\triangle ABN, \triangle ACN \in Tr_2(\Omega)$. Let $\gamma_1 := \triangle ABN \cap \partial\Omega$ and $\gamma_2 := \triangle ACN \cap \partial\Omega$. Then $\gamma_1 \in \Gamma(\angle BAN)$ and $\gamma_2 \in \Gamma(\angle CAN)$. Next, we construct the straight line segment approximations D_1E_1 and D_2E_2 of the curves γ_1 and γ_2, respectively, such that D_2 and E_1 are on AN.

3. Let $\delta_2 := \max\{\mathcal{A}(\triangle AD_1E_1 \cap \Omega), \mathcal{A}(\triangle AD_2E_2 \cap \Omega)\}$.

Once again, in order to guarantee that the error of approximation of f over the children Δ' of Δ is decreasing (at a certain rate) compared to the error of approximation of f over Δ, we want to bound the quantity $\varepsilon(\Delta')/\varepsilon\Delta$ (see more details in Remark 2.16). In particular, we aim to bound the two ratios

$$\frac{\delta_2}{\mathcal{A}(\overline{\Delta\backslash\Omega})} \text{ and } \frac{\delta_2}{\mathcal{A}(\Delta \cap \Omega)}$$

from above. The second ratio can be bounded by $5/9$ from above using Lemma 2.2 (Winternitz Theorem) (see details in Subsection 2.3.2). Hence, we consider two further subcases when $\frac{\delta_2}{\mathcal{A}(\overline{\Delta\backslash\Omega})} \leq 5/9$, and when $\frac{\delta_1}{\mathcal{A}(\overline{\Delta\backslash\Omega})} > 5/9$.

Case (1) If $\delta_2 \leq \frac{5}{9}\mathcal{A}(\overline{\Delta\backslash\Omega})$, then we proceed as follows:

3.1. Denote by M_i the centroid of $\triangle AD_iE_i \cap \Omega$, $i = 1, 2$.

3.2. Denote by F_i the midpoint of D_iE_i, $i = 1, 2$.

3.3. Δ is now divided into 14 triangles: $\triangle AM_1E_1, \triangle AD_1M_1, \triangle F_1E_1M_1$, $\triangle D_1F_1M_1, \triangle BNF_1, \triangle BD_1F_1, \triangle E_1F_1N, \triangle AM_2E_2, \triangle AD_2M_2, \triangle F_2E_2M_2$, $\triangle D_2F_2M_2, \triangle D_2F_2N, \triangle CE_2F_2$ and $\triangle CF_2N$ (see Figure 2.14).

Children: $\mathcal{C}(\Delta) \subset \big(Tr_1^*(\Omega) \cup Tr_2(\Omega) \cup Tr_3(\Omega)\big)$.

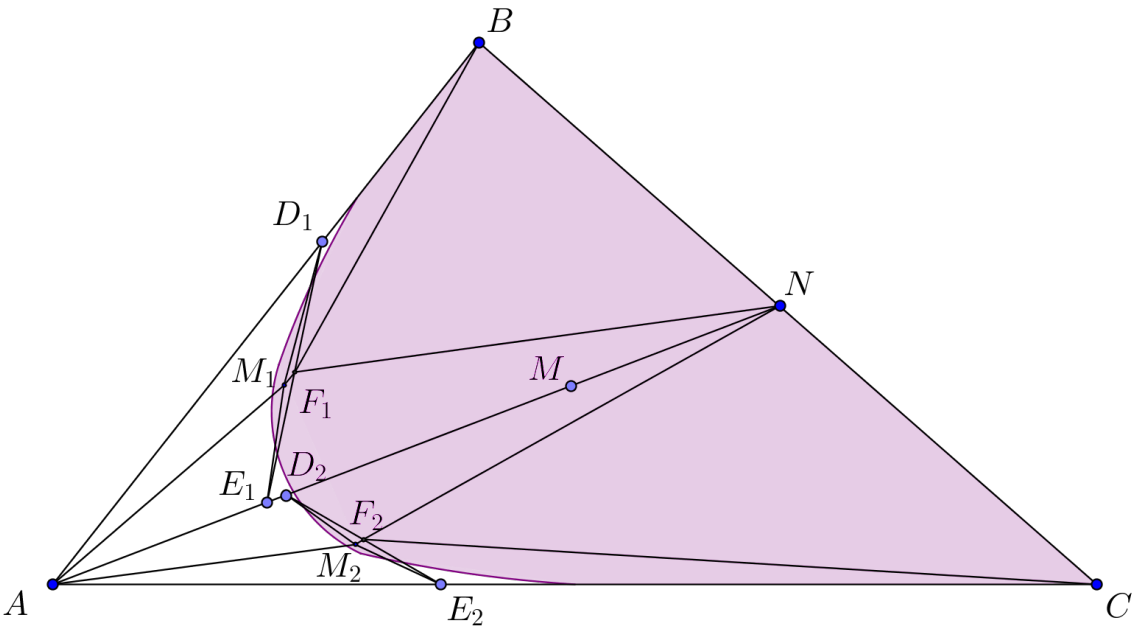


Figure 2.14: Subdivision of $\Delta \in Tr_2(\Omega)$, case (1)

Case (2) If $\delta_2 > \frac{5}{9}\mathcal{A}(\overline{\Delta\backslash\Omega})$, then

3.4. Choose a point $G \in BC$ such that $\max\{\mathcal{A}(\overline{\triangle AGB\backslash\Omega}), \mathcal{A}(\overline{\triangle AGC\backslash\Omega})\} \leq \frac{5}{9}\mathcal{A}(\overline{\Delta\backslash\Omega})$. *The procedure of finding such a point is described in Construction 2.3.*

3.5. The split consists of two triangles: $\triangle AGB$ and $\triangle AGC$ (see Figure 2.15).

Children: $\mathcal{C}(\Delta) \subset Tr_2(\Omega)$.

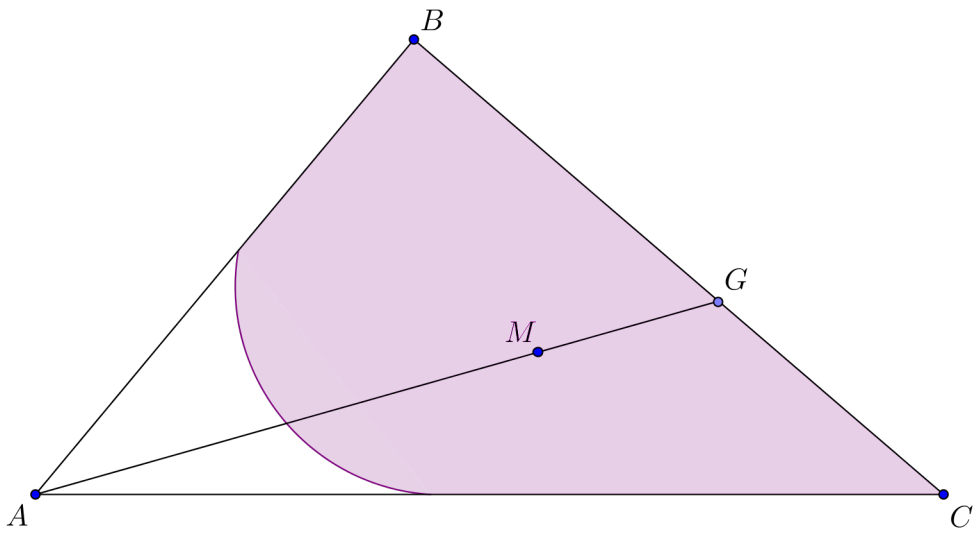


Figure 2.15: Subdivision of $\Delta \in Tr_2(\Omega)$, case (2)

Remark 2.15. *The number of children of each triangle in the algorithm is bounded by* 19.

Remark 2.16 (Cases (1) and (2) in the subdivision of $\Delta \in Tr_1^*(\Omega)$ and $\Delta \in Tr_2(\Omega)$)**.** *Both subdivision rules of $\Delta \in Tr_1^*(\Omega)$ and $\Delta \in Tr_2(\Omega)$ use the straight line segment approximation introduced in Section 2.2.4. The goal of an adaptive algorithm is to reduce the local error of approximation on the set of the triangles that are subdivided. In our setting, the error of approximation over Δ is either $\mathcal{A}(\Delta \cap \Omega)$ or $\mathcal{A}(\overline{\Delta \setminus \Omega})$ (whichever is the smaller value). That is why we need to control the ratio of the error of the straight line segment approximation and the areas $\mathcal{A}(\Delta \cap \Omega)$ or $\mathcal{A}(\overline{\Delta \setminus \Omega})$. However, we have established both positive and negative estimates on this ratio in Section 2.2.4, and thus, proposed two different subdivision rules for the cases when the ratio is less (cases (1)) and greater (cases (2)) than desired, respectively.*

Remark 2.17. *In the case, when $\Delta \cap \partial\Omega$ is a segment of a straight line or the boundary of a triangle such that $\Delta \cap \partial\Omega$ and its straight line segment approximation DE have infinitely many points of intersection, after one iteration of our algorithm the local error of approximation on Δ will become* 0.

2.3.2 Properties of the Algorithm

In this section, we want to establish the key properties of the algorithm. First, we re-state the results of Lemmas 2.9 and 2.10 in terms of the classes of triangles rather than the angular sets.

Corollary 2.2. *(a) Let Ω be convex and $\Delta = \triangle ABC \in Tr_1^*(\Omega)$ be such that $f(A) = 1$. Let also $DE = DE(\angle BAC, \Delta \cap \partial\Omega)$ be the straight line segment approximation of $\Delta \cap \partial\Omega \in \Gamma(\angle BAC)$. Then*

$$\mathcal{A}\left(\overline{\Delta \cap (\Omega \backslash \triangle ADE)}\right) \leq \frac{5}{9}\mathcal{A}(\Delta \cap \Omega). \tag{2.21}$$

(b) Let Ω be convex and $\Delta = \triangle ABC \in Tr_2(\Omega)$ be such that $f(A) = 0$. Let also $DE = DE(\angle BAC, \Delta \cap \partial\Omega)$ be the straight line segment approximation of $\Delta \cap \partial\Omega \in \Gamma(\angle BAC)$. Then

$$\mathcal{A}\left(\overline{\Delta \backslash (\Omega \cup \triangle ADE)}\right) \leq \mathcal{A}(\Delta \cap \Omega). \tag{2.22}$$

Proof. (a) Assume that $\Delta = \triangle ABC \in Tr_1^*(\Omega)$ with $f(A) = 1$ ($A \in \Omega$).
If $[\Delta \cap \partial\Omega] \in \Gamma(\angle BAC) \backslash \Gamma_0(\angle BAC)$, *i.e.*, $\Delta \cap \partial\Omega$ is a straight line segment or the boundary of some triangle $\triangle AXY$ such that the midpoints of the segments XY and $\overleftrightarrow{XY} \cap S(\angle BAC)$ coincide, then in either of these cases, the set $\Omega \cap \Delta \subset \triangle ADE$ (even though one of the points D or E may lie outside Δ), and hence,

$$\mathcal{A}\left(\overline{\Delta \cap (\Omega \backslash \triangle ADE)}\right) = 0,$$

and inequality (2.21) trivially follows.
If $\Delta \cap \partial\Omega \in \Gamma_0(\angle BAC)$, then denote by D' and E' the points of intersection of $\Delta \cap \partial\Omega$ and its straight line segment approximation $DE = DE(\angle BAC, \Delta \cap \partial\Omega)$. Note that even though one of the points D or E is not in Δ, $D', E' \in \Delta \cap \partial\Omega \subset \Delta$. In this case,

$$\overline{\Delta \cap (\Omega \backslash \triangle ADE)} = Seg(\Delta \cap \partial\Omega, D', E'),$$

and inequality (2.21) follows from Lemma 2.9.
(b) Assume that $\Delta = \triangle ABC \in Tr_2(\Omega)$ with $f(A) = 0$ ($A \notin \Omega$).

If $\Delta \cap \partial\Omega \in \Gamma(\angle BAC)\backslash\Gamma_0(\angle BAC)$, then $\Delta \cap \partial\Omega$ must be a straight line segment inside Δ. In this case, $\Delta \cap \partial\Omega = DE(\angle BAC, \Delta \cap \partial\Omega)$, and hence,

$$(\Omega \cap \Delta) \cup \triangle ADE = \Delta,$$

which implies that $\Delta\backslash(\Omega \cup \triangle ADE) = \emptyset$, and inequality (2.22) follows.
If $\Delta \cap \partial\Omega \in \Gamma_0(\angle BAC)$, then denote by P and Q the points of intersection of $\Delta \cap \partial\Omega$ and the sides AB and AC of Δ, respectively. Then by Lemma 2.10,

$$\mathcal{A}(\overline{\Phi_{\angle BAC}(\Delta \cap \partial\Omega)\backslash\triangle ADE}) \leq \mathcal{A}(\Delta \cap \partial\Omega, P, Q). \tag{2.23}$$

Note that $\Phi_{\angle BAC}(\Delta \cap \partial\Omega) = \Delta\backslash(\Omega \cup \triangle ADE)$. In addition, since $Seg(\Delta \cap \partial\Omega, P, Q) \subset \Delta \cap \Omega$,

$$\mathcal{A}(\Delta \cap \partial\Omega, P, Q) \leq \mathcal{A}(\Delta \cap \Omega),$$

which together with inequality (2.23) implies (2.22). □

Lemma 2.11. *Let Ω be convex and $\Delta \in Tr_1^*(\Omega) \cup Tr_2(\Omega)$ with $\varepsilon(\Delta) > \epsilon$ (ϵ is the precision of the adaptive triangulation algorithm). For each child Δ' of Δ obtained after applying the Main Subdivision Algorithm described in Subsection 2.3.1, we have*

$$\varepsilon(\Delta') \leq \frac{5}{9}\varepsilon(\Delta). \tag{2.24}$$

Remark 2.18. *Since $\#\mathcal{C}(\Delta)$ can be as large as* 19 *and the error of approximation over Δ is the sum of the errors over its children, inequality* (2.24) *may not seem to guarantee the decreasing order of the global error of approximation over the sequence of produced triangulations. However, combining* (2.24) *with* (2.45)*, another type of estimate on $\varepsilon(\Delta')$, allows us to prove the convergence of the global error of approximation to* 0 *in Subsection 2.3.4.*

Proof. Let $\Delta = \triangle ABC \in Tr_1^*(\Omega) \cup Tr_2(\Omega)$ such that $f(A) = 1$, if $\Delta \in Tr_1^*(\Omega)$, and $f(A) = 0$, if $\Delta \in Tr_2(\Omega)$.
Note that if $\Delta \cap \partial\Omega \in \Gamma(\angle BAC)\backslash\Gamma_0(\angle BAC)$, then by Remark 2.17,

$$\varepsilon(\Delta') = 0, \quad \text{for any } \Delta' \in \mathcal{C}(\Delta),$$

and inequality (2.24) holds. Therefore, in the rest of the proof we assume that $\Delta \cap \partial\Omega \in \Gamma_0(\angle BAC)$.

Consider two cases: $\Delta \in Tr_1^*(\Omega)$ and $\Delta \in Tr_2(\Omega)$.

Case 1: $\underline{\Delta \in Tr_1^*(\Omega)}$. Using the same notations as in the Main Subdivision Algorithm in Subsection 2.3.1 (see Figure 2.10), we recall that

$$\Delta \subset \bigcup_{i=1}^{5} \Phi_i,$$

where

$$\Phi_1 := \overline{\Delta \backslash \triangle ADE},\ \Phi_2 := \triangle DD_1D_2,\ \Phi_3 := \triangle EE_1E_2, \Phi_4 := AFD_1D_2,\ \Phi_5 := AFE_1E_2,$$

and estimate the local error over each of these five components. Since any $\triangle$ in the final split of Δ is a subset of some Φ_i,

$$\varepsilon(\triangle) \leq \max_{1\leq i\leq 5} \varepsilon(\Phi_i) \tag{2.25}$$

and in order to prove (2.24), it is sufficient to show that

$$\varepsilon(\Phi_i) \leq \frac{5}{9}\varepsilon(\Delta), \quad 1 \leq i \leq 5. \tag{2.26}$$

Corollary 2.2(a) implies that

$$\mathcal{A}(\Phi_1 \cap \Omega) \leq \frac{5}{9}\mathcal{A}(\Delta \cap \Omega). \tag{2.27}$$

Recall that F is the midpoint of DE, and hence, AF passes through the centroid of $\Delta \cap \Omega$. Therefore, by Lemma 2.2 (Winternitz Theorem)

$$\max\{\mathcal{A}(\triangle ADF \cap \Omega), \mathcal{A}(\triangle AEF \cap \Omega)\} \leq \frac{5}{9}\mathcal{A}(\Delta \cap \Omega),$$

and hence,

$$\max\{\mathcal{A}(\Phi_2 \cap \Omega), \mathcal{A}(\Phi_3 \cap \Omega)\} \leq \frac{5}{9}\mathcal{A}(\Delta \cap \Omega). \tag{2.28}$$

By Lemma 2.8, F is always in Ω, *i.e.*, $f(F) = 1$. By Lemma 2.7 and Corollary 2.1, $D, E \notin \Omega$, and so $f(D) = f(E) = 0$. Therefore, each of the triangles ADF and AEF is in $Tr_2(\Omega)$. Corollary 2.2(b) together with inequality (2.28) again implies that

$$\mathcal{A}(\overline{\Phi_i \backslash \Omega}) \leq \max\{\mathcal{A}(\Phi_2 \cap \Omega), \mathcal{A}(\Phi_3 \cap \Omega)\} \leq \frac{5}{9}\mathcal{A}(\Delta \cap \Omega), \quad i = 4, 5. \tag{2.29}$$

Recall that $\delta_1 = \max\{\mathcal{A}(\Phi_1 \cap \Omega), \mathcal{A}(\Phi_2 \cap \Omega), \mathcal{A}(\Phi_3 \cap \Omega)\}$. From inequalities (2.25), (2.27), (2.28) and (2.29), we conclude that for each of the 19 $\triangle$'s obtained by subdivision of $\Delta \in Tr_1^*(\Omega)$,

$$\varepsilon(\triangle) \leq \delta_1 \leq \frac{5}{9}\mathcal{A}(\Delta \cap \Omega). \tag{2.30}$$

Hence, if $\delta_1 \leq \frac{5}{9}\mathcal{A}(\Delta \backslash \Omega)$, then the final split of Δ consists of 19 triangles, and for each child Δ' of Δ, we clearly have inequality (2.24) of the lemma.

If $\delta_1 > \frac{5}{9}\mathcal{A}(\Delta \backslash \Omega)$, then the final split consists of two triangles $\triangle AHB$ and $\triangle AHC$ such that

$$\max\{\mathcal{A}(\overline{\triangle AHB \backslash \Omega}), \mathcal{A}(\overline{\triangle AHC \backslash \Omega})\} \leq \frac{5}{9}\mathcal{A}(\overline{\Delta \backslash \Omega}).$$

Therefore, in this case for each $\Delta' \in \mathcal{C}(\Delta)$,

$$\mathcal{A}(\Delta' \backslash \Omega) \leq \frac{5}{9}\mathcal{A}(\Delta \backslash \Omega) < \frac{5}{9}\left(\frac{9}{5}\delta_1\right) \leq \delta_1. \tag{2.31}$$

By (2.30), $\delta_1 \leq \frac{5}{9}\mathcal{A}(\Delta \cap \Omega)$. This with inequality (2.31) immediately implies

$$\varepsilon(\Delta') \leq \frac{5}{9}\varepsilon(\Delta), \quad \Delta' \in \mathcal{C}(\Delta),$$

and the proof of the lemma when $\Delta \in Tr_1^*(\Omega)$ is complete.

Case 2: $\underline{\Delta \in Tr_2(\Omega)}$. Using the same notations as in the Main Subdivision Algorithm in Subsection 2.3.1 (see Figure 2.13), we represent Δ as

$$\Delta = \triangle AD_1E_1 \cup \triangle AD_2E_2 \cup BD_1E_1N \cup CE_2D_2N,$$

and estimate the error over each of these four components.

Since AN passes through the centroid of $\Delta \cap \Omega$, by Lemma 2.2 (Winternitz theorem)

$$\max\{\mathcal{A}(\triangle ABN \cap \Omega), \mathcal{A}(\triangle ACN \cap \Omega)\} \leq \frac{5}{9}\mathcal{A}(\Delta \cap \Omega), \tag{2.32}$$

and hence,

$$\max\{\mathcal{A}(\triangle AD_1E_1 \cap \Omega), \mathcal{A}(\triangle AD_2E_2 \cap \Omega)\} \leq \frac{5}{9}\mathcal{A}(\Delta \cap \Omega). \tag{2.33}$$

Inequality (2.33) immediately implies that for each subtriangle $\triangle$ of $\triangle AD_1E_1$ and $\triangle AD_2E_2$, we have:

$$\mathcal{A}(\triangle \cap \Omega) \leq \max\{\mathcal{A}(\triangle AD_1E_1 \cap \Omega), \mathcal{A}(\triangle AD_2E_2 \cap \Omega)\} \leq \frac{5}{9}\mathcal{A}(\Delta \cap \Omega). \tag{2.34}$$

Each of the triangles ABN and ACN is in $Tr_2(\Omega)$ since $f(A) = 0$ and $f(N) = f(B) = f(C) = 1$. Therefore, Corollary 2.2(b) together with inequality (2.33) again imply that for any $\triangle$ inside quadrilaterals BD_1E_1N and CE_2D_2N,

$$\mathcal{A}(\triangle\backslash\Omega) \leq \max\{\mathcal{A}(\triangle AD_1E_1 \cap \Omega), \mathcal{A}(\triangle AD_2E_2 \cap \Omega)\} \leq \frac{5}{9}\mathcal{A}(\Delta \cap \Omega). \tag{2.35}$$

Recall that $\delta_2 = \max\{\mathcal{A}(\triangle AD_1E_1 \cap \Omega), \mathcal{A}(\triangle AD_2E_2 \cap \Omega)\}$. From inequalities (2.34) and (2.35), we conclude that for each of 14 $\triangle$'s mentioned in part 3.3 of the Main Subdivision Algorithm for $\Delta \in Tr_2(\Omega)$,

$$\varepsilon(\triangle) \leq \delta_2 \leq \frac{5}{9}\mathcal{A}(\Delta \cap \Omega). \tag{2.36}$$

Hence, if $\delta_2 \leq \frac{5}{9}\mathcal{A}(\Delta\backslash\Omega)$, then the final split of Δ consists of the 14 triangles, and for every child Δ' of Δ, we clearly have inequality (2.24) of the lemma.
If $\delta_2 > \frac{5}{9}\mathcal{A}(\Delta\backslash\Omega)$, then the final split consists of two triangles: $\triangle ABG$ and $\triangle ACG$ such that

$$\max\{\mathcal{A}(\overline{\triangle ABG\backslash\Omega}), \mathcal{A}(\overline{\triangle ACG\backslash\Omega})\} \leq \frac{5}{9}\mathcal{A}(\overline{\Delta\backslash\Omega}).$$

Therefore, in this case for each $\Delta' \in \mathcal{C}(\Delta)$,

$$\mathcal{A}(\Delta'\backslash\Omega) \leq \frac{5}{9}\mathcal{A}(\Delta\backslash\Omega) < \frac{5}{9}\left(\frac{9}{5}\delta_2\right) \leq \delta_2. \tag{2.37}$$

By (2.36), $\delta_2 \leq \frac{5}{9}\mathcal{A}(\Delta \cap \Omega)$. This with inequality (2.37) immediately implies

$$\varepsilon(\Delta') \leq \frac{5}{9}\varepsilon(\Delta), \quad \Delta' \in \mathcal{C}(\Delta),$$

and this completes the proof of the lemma. □

According to the Main Subdivision Algorithm, for any $\Delta \in Tr_i(\Omega)$, $i = 1, 2$,

$$\mathcal{C}(\Delta) \subset \big[Tr_1^*(\Omega) \cup Tr_2(\Omega) \cup Tr_3(\Omega)\big],$$

and only children of $\Delta \in Tr_0(\Omega)$ may belong to $Tr_1(\Omega)$, a class wider than $Tr_1^*(\Omega)$ (in the case when Δ is not subdivided (*i.e.*, $\varepsilon(\Delta) \leq \epsilon$), $\mathcal{C}(\Delta) = \emptyset$). However, it is easy to see that for any Δ,

$$\mathcal{C}^{(2)}(\Delta) \subset \big[Tr_1^*(\Omega) \cup Tr_2(\Omega) \cup Tr_3(\Omega)\big],$$

where $\mathcal{C}^{(2)}(\Delta)$ denotes the set of children of Δ of second generation (see Definition 2.1), and, once again, $\mathcal{C}^{(2)}(\Delta)$ can possibly be $\emptyset$. Hence, after performing the first two iterations of the Main Subdivision Algorithm on the initial triangulation, the input triangles that actually require a further split will always be from $Tr_1^*(\Omega) \cup Tr_2(\Omega)$. Therefore, the following corollary from Lemma 2.11 applies:

Corollary 2.3. *Let $\mathcal{D}_0$ be an initial triangulation of convex $\Omega \subset [0,1]^2$ and $\{\mathcal{D}_i\}_{i=1}^N$ be the sequence of triangulations produced by the adaptive triangulation algorithm with the subdivision rule introduced in Subsection 2.3.1 and a precision $\epsilon > 0$ ($0 \leq N \leq \infty$). Then for any $\Delta \in \cup_{1\leq i\leq N}\mathcal{D}_i$ with $\varepsilon(\Delta) > \epsilon$,*

$$\varepsilon(\Delta') \leq 5/9\varepsilon(\Delta), \quad \Delta' \in \mathcal{C}(\Delta). \tag{2.38}$$

Remark 2.19. *Corollary 2.3 guarantees termination of our adaptive triangulation algorithm for any initial precision $\epsilon > 0$, i.e., N is finite and for every Δ in the final triangulation $\mathcal{D}_N$, $\varepsilon(\Delta) \leq \epsilon$.*

In addition to the decreasing rate of $\varepsilon(\Delta)$ established in Lemma 2.11 and Corollary 2.3, we can also provide the estimate of $\varepsilon(\Delta)$ in terms of some geometric characteristics of $\Delta \cap \partial\Omega$.

Definition 2.7. *Let $\Delta \in Tr_1^*(\Omega) \cup Tr_2(\Omega)$ and $\gamma = \Delta \cap \partial\Omega$, for some convex $\Omega \subset [0,1]^2$. Suppose that γ intersects the sides of Δ at points P and Q. Then,*

$$\mathcal{A}(\gamma, \Delta) := \mathcal{A}\left(Seg\left(\gamma, P, Q\right)\right), \qquad \mathcal{L}(\gamma, \Delta) := |\gamma(P, Q)|.$$

Corollary 2.4. *Let $\Delta \in Tr_1^*(\Omega) \cup Tr_2(\Omega)$ and $\gamma = \Delta \cap \partial\Omega$, for some convex $\Omega \subset [0,1]^2$. If $\varepsilon(\Delta) > \epsilon$ and Δ is subdivided using the Main Subdivision Algorithm introduced in Subsection 2.3.1, then*

$$\varepsilon(\Delta') \leq \mathcal{A}(\gamma, \Delta), \quad \Delta' \in \mathcal{C}(\Delta). \tag{2.39}$$

Proof. Let $\Delta \in Tr_1^*(\Omega)$ or $\Delta \in Tr_2(\Omega)$, and label the vertices of Δ with A, B, C so that $f(A) = 1$ if $\Delta \in Tr_1^*(\Omega)$ and $f(A) = 0$ if $\Delta \in Tr_2(\Omega)$. Furthermore, let $\gamma = \Delta \cap \partial\Omega$ be a curve intersecting AB and AC at P and Q, respectively.

Using the same notations as in the proof of Lemma 2.11 and in Figures 2.10, 2.14, we have

$$\Phi_1 \cap \Omega,\ \Phi_2 \cap \Omega,\ \Phi_3 \cap \Omega \subset Seg(\gamma, P, Q), \quad \text{if } \Delta \in Tr_1^*(\Omega),$$

and

$$\triangle AD_1E_1 \cap \Omega,\ \triangle AD_2E_2 \cap \Omega \subset Seg(\gamma, P, Q), \quad \text{if } \Delta \in Tr_2(\Omega).$$

The definition of δ_i's implies that

$$\delta_1 \leq \mathcal{A}(\gamma, \Delta),\ \Delta \in Tr_1^*(\Omega), \qquad \text{and} \qquad \delta_2 \leq \mathcal{A}(\gamma, \Delta),\ \Delta \in Tr_2(\Omega),$$

which together with inequalities (2.30),(2.31),(2.36) and (2.37) implies (2.39). □

As implicitly mentioned in Remark 2.19, Corollary 2.3 actually implies that the local error approaches 0. Now using Remark 2.15 and Corollary 2.4, we want to control the total number of triangles in the triangulation.

2.3.3 Further Assumptions on $\partial\Omega$

We recall that, in our setting, $\partial\Omega$ has to be a piecewise-smooth curve (Ω can be a polygon, and thus does not have to have a smooth boundary everywhere). More precisely, for a convex Ω, its boundary $\partial\Omega$ must satisfy the following assumptions:

- $\partial\Omega = \bigcup_{i=1}^{q} \gamma_k$;
- Choosing a suitable local coordinate system, we represent each γ_i as the graph of $y = g_i(x), a_i \leq x \leq b_i,\ g_i \in C^{(2)}[a_i, b_i]$;

In other words, we can always choose q points K_i on the boundary $\partial\Omega$ such that $\partial\Omega(K_i, K_{i+1})$ is a $C^{(2)}$-curve for any i, $1 \leq i \leq q$ ($K_{q+1} := K_1$). In this case, for any curve $\widetilde{\gamma} \subset \partial\Omega$ that does not contain any of the points K_i in its interior, this portion of the boundary of Ω is the graph of some $y = g(x)$, $x \in [a, b]$, with $g \in C^{(2)}[a, b]$, in

a suitable local coordinate system. Let $A := (a, g(a))$ and $B := (b, g(b))$. Denoting by $l(x)$ the line $\overleftrightarrow{AB}$ and using the notation introduced in Notation 2.2,

$$\mathcal{A}(\widetilde{\gamma}, A, B) = \left| \int_a^b g(x) - l(x)\, dx \right| \leq \|g - l\|_{C[a,b]}(b-a) = |g(\xi) - l(\xi)|(b-a), \quad (2.40)$$

where $\xi = \text{argmax}_{x\in[a,b]}|g(x) - l(x)|$. Taking into account that $g'(\xi) - l'(\xi) = 0$ and $l'' \equiv 0$, we get from

$$0 = g(a) - l(a) = [g(\xi) - l(\xi)] + [(g'(\xi) - l'(\xi))(a-\xi)] + \frac{g''(\eta)}{2}(a-\xi)^2,$$

for some $\eta \in (a, \xi)$, that

$$|g(\xi) - l(\xi)| \leq \frac{\|g''\|_{C[a,b]}}{2}(b-a)^2 = c_0(b-a)^2, \quad (2.41)$$

for some constant c_0 depending on $\partial\Omega$ only. Inequalities (2.40) and (2.41) imply that

$$\mathcal{A}(\widetilde{\gamma}, A, B) \leq c_0(b-a)^3 \leq c_0|\widetilde{\gamma}(A, B)|^3. \quad (2.42)$$

Remark 2.20. *Note that if $\widetilde{\gamma} = \widetilde{\gamma}(A, B) \subset \partial\Omega$ contains some of the points K_i, (i.e., $\widetilde{\gamma}$ is not necessarily a $C^{(2)}$-curve), the Isoperimetric Inequality*

$$A \leq \frac{L^2}{4\pi},$$

where L is the length of some closed curve and A is the area of the planar region that it encloses, implies

$$\mathcal{A}(\widetilde{\gamma}, A, B) \leq \frac{(|AB| + |\widetilde{\gamma}(A, B)|)^2}{4\pi} \leq \frac{|\widetilde{\gamma}(A, B)|^2}{\pi}. \quad (2.43)$$

Under these assumptions on the boundary of Ω, we may combine inequalities (2.38), (2.39),(2.42) and (2.43) into the following theorem stating the main properties of the algorithm:

Theorem 2.1. *Let $\Omega \subset [0,1]^2$ be a convex set such that the boundary of Ω is a $C^{(2)}$-curve except at the set of points*

$$\mathcal{K} := \{K_1, K_2, \ldots, K_q\},$$

and let $\mathcal{D}_i$, $1 \leq i \leq N$, *be the partitions produced by the adaptive triangulation algorithm with a precision* $\epsilon > 0$ *and the subdivision rule introduced in Subsection 2.3.1. Then for any* $\Delta \in \cup_{2\leq i\leq N}\mathcal{D}_i$ *and* $\Delta' \in \mathcal{C}(\Delta)$,

$$\varepsilon(\Delta') \leq 5/9\varepsilon(\Delta) \tag{2.44}$$

and, with $\gamma := \partial\Omega$ *and* $\mathcal{L}$ *as defined in Notation 2.2,*

$$\varepsilon(\Delta') \leq \begin{cases} c_0(\mathcal{L}(\gamma,\Delta))^3, & \text{if } \mathcal{K} \cap \operatorname{int}(\Delta) = \emptyset, \\ (\mathcal{L}(\gamma,\Delta))^2, & \text{if } \mathcal{K} \cap \operatorname{int}(\Delta) \neq \emptyset. \end{cases} \tag{2.45}$$

Remark 2.21. *In the next section, we relabel* $\mathcal{D}_i$*'s so that* $\mathcal{D}_2$ *becomes the initial triangulation, and after such relabeling, we can assume that inequalities* (2.44) *and* (2.45) *hold for any* $\Delta \in (\mathcal{D}_0 \cup \mathcal{D}_1)$ *with* $\varepsilon(\Delta) > \epsilon$ *as well.*

2.3.4 Convergence of the Algorithm

In the following theorem, we establish convergence of our adaptive triangulation algorithm along with some rates of its convergence.

Theorem 2.2. *Let* $f(x) = \chi_\Omega(x)$, $x \in [0,1]^2$, *be the characteristic function of a convex set* $\Omega \subset [0,1]^2$ *with a piecewise-smooth boundary* $\gamma := \partial\Omega$, *and let* $\delta := \max_{\Delta\in\mathcal{D}_0} \varepsilon(\Delta)$ *be the maximum of the local error over the initial triangulation* $\mathcal{D}_0$. *Then the adaptive triangulation algorithm with the subdivision rule introduced in Subsection 2.3.1 and a precision* $\epsilon \in (0,\delta)$ *produces the final triangulation* $\mathcal{D}$ *with*

$$\#\mathcal{D} \leq C\epsilon^{-1/3}\ln(1/\epsilon), \tag{2.46}$$

and

$$\sigma_1(f,\mathcal{D})_1 \leq C\epsilon^{2/3}\ln(1/\epsilon), \tag{2.47}$$

where the constant C *does not depend on* ϵ *and* $\sigma_1(f,\mathcal{D}_1)$ *is as defined by* (2.2).

Proof. Let the set Ω and precision ϵ be now fixed. Assume that the conditions of the theorem are satisfied and that our initial partition $\mathcal{D}_0$ has k_0 triangles. For simplicity of notation, let $\alpha := 5/9$, $\sigma := 19$, and since $\gamma = \partial\Omega$ is now fixed as well, let $\mathcal{L}_\Delta := \mathcal{L}(\gamma, \Delta)$.

Due to inequality (2.44), our adaptive algorithm will terminate, and let $\mathcal{D}_N$ be the final partition. Then the global error of approximation over $\mathcal{D}_N$ (see Definition 2.2) can be bounded as follows:

$$\sigma_1(f, \mathcal{D}_N)_1 = \sum_{\Delta \in \mathcal{D}_N} \varepsilon(\Delta) \leq \epsilon \# \mathcal{D}_N, \tag{2.48}$$

and we now want to estimate the number of triangles in $\mathcal{D}_N$ using inequalities (2.44) and (2.45).

Let $\mathcal{P}(\mathcal{D}_N) := \{\mathcal{P}(\Delta) : \Delta \in \mathcal{D}_N\}$. Following the methodology in [9], we write these 'parental' sets as follows

$$\mathcal{P}(\mathcal{D}_N) = \bigcup_{j \in \mathbb{Z}} \Pi_j, \quad \text{where } \Pi_j := \{\Delta \in \mathcal{P}(\mathcal{D}_N) : \alpha^{j+1} \leq \varepsilon(\Delta) < \alpha^j\}. \tag{2.49}$$

Due to Remark 2.1 and inequality (2.44), all the triangles in each Π_j are disjoint. Since each $\Delta \in \mathcal{P}(\mathcal{D}_N)$ was subdivided before obtaining the final partition $\mathcal{D}_N$, $\epsilon < \varepsilon(\Delta)$, and so we get a restriction on the index j:

$$\epsilon < \alpha^j,$$

which is equivalent to $j < \log_\alpha \epsilon$ since $0 < \alpha < 1$.
Another restriction on j can be derived from the fact that the local error may only decrease and cannot exceed the maximum of the local error over the initial partition. For every $\Delta \in \mathcal{P}(\mathcal{D}_N)$,

$$\varepsilon(\Delta) \leq \max_{\Delta_0 \in \mathcal{D}_0} \varepsilon(\Delta_0) = \delta \;\Rightarrow\; \alpha^{j+1} \leq \delta \;\Rightarrow\; j \geq \log_\alpha \delta - 1.$$

The above inequalities imply that

$$\#\mathcal{P}(\mathcal{D}_N) = \sum_{j=j_0}^{j^*} \#\Pi_j,$$

where $j_0 := \lceil \log_\alpha \delta \rceil - 1$ and $j^* = \lceil \log_\alpha \epsilon \rceil - 1$. Taking into account that each parent from $\mathcal{P}(\mathcal{D}_N)$ may have no more than σ children and that some of the cells from the final partition $\mathcal{D}_N$ may be from the initial partition $\mathcal{D}_0$ and not have any parents, we estimate

$$\#\mathcal{D}_N \leq \sigma \#\mathcal{P}(\mathcal{D}_N) + k_0 = \sigma \sum_{j=j_0}^{j^*} \#\Pi_j + k_0. \tag{2.50}$$

Now, we want to estimate the number of cells in each Π_j by going one more generation up in our forest F_N. Recall that $\mathcal{K}$ is the collection of 'singularity' points on $\partial\Omega$. Then, any Δ in the set Π_j either has a parent or does not have one (which means $\Delta \in \mathcal{D}_0$). For every j, $j_0 \leq j \leq j^*$, we write Π_j as the following disjoint union

$$\Pi_j = \Pi_j^{\mathcal{S}} \cup \Pi_j^{\mathcal{NS}} \cup (\Pi_j \cap \mathcal{D}_0),$$

where

$$\Pi_j^{\mathcal{S}} := \{\Delta \in \Pi_j : \mathrm{int}(\mathcal{P}(\Delta)) \cap \mathcal{K} = \emptyset\},$$
$$\Pi_j^{\mathcal{NS}} := \{\Delta \in \Pi_j : \mathrm{int}(\mathcal{P}(\Delta)) \cap \mathcal{K} \neq \emptyset\}.$$

For every $\Delta \in \Pi_j^{\mathcal{S}}$, inequality (2.45) implies that $\varepsilon(\Delta) \leq c_0 \mathcal{L}^3_{\mathcal{P}(\Delta)}$, and hence,

$$1 \leq \alpha^{-j-1}\varepsilon(\Delta) \leq c_0 \alpha^{-j-1} \mathcal{L}^3_{\mathcal{P}(\Delta)} \Rightarrow 1 \leq (c_0)^{1/3} \alpha^{(-j-1)/3} \mathcal{L}_{\mathcal{P}(\Delta)}.$$

Therefore,

$$\begin{aligned} \#\Pi_j^{\mathcal{S}} \leq c_0^{1/3} \alpha^{(-j-1)/3} \sum_{\Delta \in \Pi_j^{\mathcal{S}}} \mathcal{L}_{\mathcal{P}(\Delta)} &\leq c_0^{1/3} \alpha^{(-j-1)/3} \sum_{\Delta \in \Pi_j} \mathcal{L}_{\mathcal{P}(\Delta)} \\ &\leq c_0^{1/3} \alpha^{(-j-1)/3} \sigma \sum_{\Delta \in \mathcal{P}(\Pi_j)} \mathcal{L}_\Delta. \end{aligned} \tag{2.51}$$

Note that the elements of the set $\mathcal{P}(\Pi_j) = \{\mathcal{P}(\Delta) : \Delta \in \Pi_j\}$ do not have to be disjoint. However, as mentioned in Remark 2.1, for any two distinct triangular cells $\Delta_1, \Delta_2 \in \mathcal{F}_N$ with a common interior point, $\Delta_1 \subset \Delta_2$ or $\Delta_2 \subset \Delta_1$. Due to this property, we can find elements $\Delta_k^* \in \mathcal{P}(\Pi_j)$, $1 \leq k \leq \nu$, such that for every $\Delta \in \mathcal{P}(\Pi_j)$,

$$\Delta \subset \Delta_k^*, \quad \text{for some } 1 \leq k \leq \nu,$$

and

$$\operatorname{int}(\Delta_k^* \cap \Delta_m^*) = \emptyset,\ k \neq m.$$

Hence, we can write $\mathcal{P}(\Pi_j)$ as the disjoint union

$$\mathcal{P}(\Pi_j) = \bigcup_{k=1}^{\nu} \{\Delta \in \mathcal{P}(\Pi_j) : \Delta \subset \Delta_k^*\} =: \bigcup_{k=1}^{\nu} X_k, \quad X_k \cap X_m = \emptyset,\ k \neq m.$$

From a geometric point of view, the union of all triangles in $\mathcal{P}(\Pi_j)$ is equal to the union of the Δ_k^*'s. Each X_k is a subset of triangles of the tree that is a subgraph of $\mathcal{F}_N$ and has Δ_k^* as its root. Also, by the definition of j_0,

$$\alpha^{j-j_0}\varepsilon(\Delta_k^*) \leq \alpha^j \alpha^{-j_0}\delta < \alpha^j, \tag{2.52}$$

and for each $\Delta \in \mathcal{P}(\Pi_j)$,

$$\varepsilon(\Delta) \geq \alpha^j, \tag{2.53}$$

which together with inequality (2.44) implies that the elements of $\mathcal{P}(\Pi_j)$ may appear among at most the first $j - j_0 - 1$ generations of children of Δ_k^* (since otherwise inequalities (2.52) and (2.53) must hold simultaneously). Hence,

$$\begin{aligned}\sum_{\Delta\in\mathcal{P}(\Pi_j)} \mathcal{L}_\Delta &= \sum_{k=1}^{\nu} \sum_{\Delta\in X_k} \mathcal{L}_\Delta = \sum_{k=1}^{\nu} \sum_{m=0}^{j-j_0-1} \sum_{\Delta\in\left[X_k\cap\mathcal{C}^{(m)}(\Delta_k^*)\right]} \mathcal{L}_\Delta \\ &\leq \sum_{k=1}^{\nu} \sum_{m=0}^{j-j_0-1} \mathcal{L}_{\Delta_k^*} = (j-j_0)\sum_{k=1}^{\nu} \mathcal{L}_{\Delta_k^*} \leq (j-j_0)\mathcal{L}_0,\end{aligned}$$

where $\mathcal{L}_0$ is the length of the curve $\partial\Omega$, and we used the fact that for every Δ and $m \geq 0$, the disjoint union $\bigcup_{\Delta'\in\mathcal{C}^{(m)}(\Delta)} \Delta'$ is always contained in Δ. The last chain of inequalities together with (2.51) implies that

$$\#\Pi_j^{\mathcal{S}} \leq c_0^{1/3}\sigma(j-j_0)\alpha^{(-j-1)/3}\mathcal{L}_0. \tag{2.54}$$

Now we estimate $\#\mathcal{P}(\Pi_j^{\mathcal{NS}})$. Let $Y_i = \{\Delta \in \mathcal{P}(\Pi_j^{\mathcal{NS}}) : K_i \in \operatorname{int}(\Delta)\}$, $1 \leq i \leq q$. It is easy to see that each Y_i is a subset of some X_k (with corresponding root Δ_k^*). Moreover, since all triangles in Y_i have a common interior point K_i, each Y_i is just a

collection of some children of Δ_k^* of *different* generations, and since it is contained in some X_k, the number of triangles in this collection does not exceed $j - j_0$. Taking into account that $\mathcal{P}(\Pi_j^{\mathcal{NS}}) = \bigcup_{i=1}^q Y_i$ (not necessarily a disjoint union), we proceed to the estimate

$$\#\Pi_j^{\mathcal{NS}} \le \sigma \#\mathcal{P}(\Pi_j^{\mathcal{NS}}) \le \sigma \sum_{i=1}^{q} \#Y_i \le \sigma q(j - j_0). \tag{2.55}$$

We now come back to the estimate (2.50) applying inequalities (2.54) and (2.55)

$$\begin{aligned}
\#\mathcal{D}_N &\le \sigma \sum_{j=j_0}^{j^*} \#\Pi_j + k_0 \\
&\le \sigma \sum_{j=j_0}^{j^*} \left[\#\Pi_j^{\mathcal{S}} + \#\Pi_j^{\mathcal{NS}} + \#(\Pi_j \cap \mathcal{D}_0)\right] + k_0 \\
&\le \sigma \sum_{j=j_0}^{j^*} \left[c_0^{1/3}\sigma(j - j_0)\alpha^{(-j-1)/3}\mathcal{L}_0 + \sigma q(j - j_0) + k_0\right] + k_0 \\
&\le k_0(\sigma(j^* - j_0 + 1) + 1) + \sigma^2 \sum_{j=0}^{j^*-j_0} \left[j\left(c_0^{1/3}\alpha^{(-j-j_0-1)/3}\mathcal{L}_0 + q\right)\right] \\
&\le k_0\sigma(j^* - j_0 + 1) + k_0 + \frac{\sigma^2 q(j^* - j_0)(j^* - j_0 + 1)}{2} + \frac{\sigma^2 c_0^{1/3}\mathcal{L}_0}{\alpha^{(j_0+1)/3}} \sum_{j=1}^{j^*-j_0} j\alpha^{-j/3} \\
&\le k_0\sigma(j^* - j_0 + 1) + k_0 + \frac{\sigma^2 q(j^* - j_0)(j^* - j_0 + 1)}{2} \\
&\quad + \frac{\sigma^2 c_0^{1/3}\mathcal{L}_0}{\alpha^{(j_0+1)/3}} \int_1^{j^*-j_0+1} x\alpha^{-x/3}\, dx.
\end{aligned}$$

Recalling that $j_0 = \lceil \log_\alpha \delta \rceil - 1$ and $j^* = \lceil \log_\alpha \epsilon \rceil - 1$, we get

$$\#\mathcal{D}_N \le c_1 + c_2 j^* + c_3 (j^*)^2 + c_4 j^* \alpha^{-(j^*+1)/3} \le c_1 + c_2 \log_\alpha \epsilon + c_3 (\log_\alpha \epsilon)^2 + c_4 (\log_\alpha \epsilon)\epsilon^{-1/3},$$

where $c_1 = c_1(\sigma, k_0)$, $c_2 = c_2(\alpha, \delta, \sigma, k_0)$ $c_3 = c_3(\alpha, \delta, \sigma, q)$ and $c_4 = c_4(\alpha, \delta, \sigma, \Omega)$. Therefore, for $\epsilon \in (0, \delta)$, by inequality (2.48)

$$\#\mathcal{D}_N \le C\epsilon^{-1/3} \log_\alpha \epsilon, \qquad \sigma_1(f, \mathcal{D}_N)_1 \le C\epsilon^{2/3} \log_\alpha \epsilon, \tag{2.56}$$

where the constant C does not depend on ϵ, and the statement of the theorem follows. □

Remark 2.22. *Note that if $\epsilon \geq \delta$ (keeping the notations of Theorem 2.2 and its proof), no subdivision is necessary and*

$$\#\mathcal{D}_N = \#\mathcal{D}_0, \qquad \sigma_1(f, \mathcal{D}_N)_1 \leq \delta \#\mathcal{D}_0.$$

If we take $\epsilon = n^{-3}$ in Theorem 2.2, then inequalities (2.46) and (2.47) provide the following result:

Theorem 2.3. *Let $f(x) = \chi_\Omega(x)$, $x \in [0,1]^2$, be the characteristic function of a convex set $\Omega \subset [0,1]^2$ with a piecewise-smooth boundary, and let $\delta = \max_{\Delta_0 \in \mathcal{D}_0} \varepsilon(\Delta_0)$ be the maximum of the local error over the initial triangulation $\mathcal{D}_0$. Then for any $n > \delta^{-1/3}$, the adaptive triangulation algorithm with the subdivision rule introduced in Subsection 2.3.1 and precision $\epsilon := n^{-3}$ produces the final triangulation $\mathcal{D}$ with*

$$\#\mathcal{D} \leq Cn \ln n$$

and the global error $\sigma_1(f, \mathcal{D})_1$ of the piecewise-constant approximation of f on the partition $\mathcal{D}$ is bounded as follows

$$\sigma_1(f, \mathcal{D})_1 \leq \frac{C \ln n}{n^2}$$

(C is a positive constant independent of n).

Also, if we take

$$\epsilon^{-1/3} \log_\alpha \epsilon = n \tag{2.57}$$

in (2.56), then we obtain the following corollary:

Corollary 2.5. *Under the settings of Theorem 2.3, for any $n > \delta^{-1/3}$, the adaptive triangulation algorithm with the subdivision rule introduced in Subsection 2.3.1 and precision ϵ that satisfies (2.57) produces the final triangulation $\mathcal{D}$ with*

$$\#\mathcal{D} \leq Cn, \qquad \sigma_1(f, \mathcal{D})_1 \leq \frac{C(\ln n)^3}{n^2}$$

Proof. Indeed, if $\epsilon^{-1/3} \log_\alpha \epsilon = n$, then inequality (2.56) implies that

$$\sigma_1(f, \mathcal{D})_1 \leq C\epsilon^{2/3} \log_\alpha \epsilon = C \frac{\log_\alpha^3 \epsilon}{\epsilon^{-2/3} \log_\alpha^2 \epsilon} = \tilde{C} \frac{\ln^3 n}{n^2},$$

where the last equality holds since $-1/3 \log_\alpha \varepsilon + \log_\alpha(\log_\alpha \varepsilon) = \log_\alpha n$. □

Comparing the result of Corollary 2.5 and inequality (1.4), we can conclude that our algorithm provides a nearly optimal error of approximation of the characteristic functions of convex domains over a triangulation with n triangles.

Chapter 3

Conclusions

In this doctoral thesis, we introduce a new adaptive method that has the potential to be used in Image Coding. In particular, given a characteristic function of some bounded convex domain (a so-called cartoon image), our algorithm constructs a hierarchial sequence of triangulations that adapt to the local properties of the function. In case of convex domains with piecewise-smooth boundary, this approximation method implies a 'theoretically correct' rate of convergence and already outperforms the well-known wavelet methods. Moreover, our approach can also be extended to broader classes of not necessarily convex domains, since the smooth boundary of any non-convex set is locally convex. All of this together with an extension of this adaptive method to multivariate piecewise polynomial approximation is a subject for future investigation.

Bibliography

[1] P. Binev, W. Dahmen, and R. DeVore, *Adaptive finite element methods with convergence rates*, Numer. Math. **97** (2004), no. 2, 219–268.

[2] P. Binev, W. Dahmen, R. DeVore, and P. Petrushev, *Approximation classes for adaptive methods*, Serdica Math. J. **28** (2002), no. 4, 391–416. Dedicated to the memory of Vassil Popov on the occasion of his 60th birthday.

[3] W. Blaschke, *Vorlesungen über Differentialgeometrie. II, Affine Differentialgeometrie*, Springer, Berlin, 1923.

[4] E. J. Candès and D. L. Donoho, *Curvelets and curvilinear integrals*, J. Approx. Theory **113** (2001), no. 1, 59–90.

[5] ———, *New tight frames of curvelets and optimal representations of objects with piecewise C^2 singularities*, Comm. Pure Appl. Math. **57** (2004), no. 2, 219–266.

[6] A. Cohen, R. DeVore, P. Petrushev, and H. Xu, *Nonlinear approximation and the space* $\mathrm{BV}(\mathbf{R}^2)$, Amer. J. Math. **121** (1999), no. 3, 587–628.

[7] A. Cohen, N. Dyn, F. Hecht, and J. M. Mirebeau, *Adaptive multiresolution analysis based on anisotropic triangulations*, Math. Comp. **81** (2012), no. 278, 789–810.

[8] I. Daubechies, *Ten lectures on wavelets*, CBMS-NSF Regional Conference Series in Applied Mathematics, vol. 61, Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 1992.

[9] R. DeVore, K. Kopotun, and B. Popov, *Adaptive Approximation (TAMU, November/December 2008): Notes* (January 25, 2009).

[10] D. L. Donoho, *Wedgelets: nearly minimax estimation of edges*, Ann. Statist. **27** (1999), no. 3, 859–897.

[11] ———, *Sparse components of images and optimal atomic decompositions*, Constr. Approx. **17** (2001), no. 3, 353–382.

[12] E. Ehrhart, *Une généralisation du théorème de Minkowski (French)*, C. R. Acad. Sci. Paris **240** (1955), 483–485.

[13] C. S. Güntürk, *Wavelets, Approximation Theory, and Signal Processing*, Notes, *Scribe: Evan Chou* (Fall 2010).

[14] A. Haar, *Zur Theorie der orthogonalen Funktionensysteme*, Math. Ann. **69** (1910), no. 3, 331–371.

[15] D. Huffman, *A method for the construction of minimum redundancy codes*, Proc. IRE 40 **9** (1952), 1098-1101.

[16] R. A. Johnson, *Advanced Euclidean Geometry*, Dover, 2007.

[17] B. S. Kashin, *Approximation properties of complete orthonormal systems*, Trudy Mat. Inst. Steklov. **172** (1985), 187–191, 353. Studies in the theory of functions of several real variables and the approximation of functions.

[18] R. Kazinnik, S. Dekel, and N. Dyn, *Low bit-rate image coding using adaptive geometric piecewise polynomial approximation*, IEEE Transactions on Image Processing **6** (September, 2007), no. 9.

[19] M. Kunt, A. Ikonomopoulos, and M. Kocher, *Second generation image coding*, Proc. IEEE **73** (1985), no. 4, 549-574.

[20] R. Larson and B. H. Edwards, *Calculus of a Single Variable, Ninth Edition*, Brooks/Cole, 2010.

[21] M. A. Lavrent'ev and L. A. Lyusternik, *Fundamentals of the Calculus of Variations (in Russian)*, Vol. 1, Part II, Moscow, 1935.

[22] S. Mallat and G. Peyré, *A review of bandlet methods for geometrical image representation*, Numer. Algorithms **44** (2007), no. 3, 205–234.

[23] Y. Meyer, *Wavelets and operators*, Cambridge Studies in Advanced Mathematics, vol. 37, Cambridge University Press, Cambridge, 1992. Translated from the 1990 French original by D. H. Salinger.

[24] J. M. Mirebeau, *Adaptive and anisotropic finite element approximation: Theory and algorithms*, PhD Thesis, 2010.

[25] B. H. Neumann, *On an invariant of plane regions and mass distributions*, J. London Math. Soc. **20** (1945), 226–237.

[26] D. J. Newman, *Partitioning of areas by straight lines*, Notices Amer. Math. Soc **5** (1958), 510.

[27] I. Ya. Novikov, V. Yu. Protasov, and Skopina M. A., *Wavelet theory*, Vol. 239, American Mathematical Society, 2011.

[28] M. Reid, R. Millar, and Black N., *Second-Generation Image Coding: An Overview*, ACM Computing Surveys **29** (March 1997), 3–29.

[29] ———, *A comparison of first generation image coding techniques applied to the same magnetic resonance image*, Innovation et Technologie en Biologie et Medicine 15 **4, 512** (1994).

[30] J. Sampo, *Some remarks on convergence of curvelet transform of piecewise smooth functions*, Appl. Comput. Harmon. Anal. **34** (2013), no. 2, 324–326.

[31] R. Shukla, P. L. Dragotti, M. N. Do, and M. Vetterli, *Rate-distortion optimized tree-structured compression algorithms for piecewise polynomial images*, IEEE Trans. Image Process. **14** (2005), no. 3, 343–359.

[32] D. Taubman and M. Marcellin, *JPEG2000: Image Compression Fundamentals, Standards, and Practice*, Kluwer International Series in Engineering and Computer Science, 2001.

[33] I. M. Yaglom and V.G. Boltyanski, *Convex Figures (in Russian)*, Vol. 4, State Technical-Theoretical Literature Publisher, Moscow, 1951.

[34] M. Wakin, J. Romberg, H. Choi, and R. Baraniuk, *Geometric methods for wavelet-based image compression (conference proceedings)*, SPIE Wavelets X, San Diego, August, 2003.

[35] T. Welch, *A technique for high performance data compression*, IEEE Computing **17** (1984), no. 6, 8–19.

Appendix A

Matlab Code with Implementation

Auxiliary Functions

Boundaries for a triangular set of pixels

```
%Boundaries of a triangle
function[low,upp,lt,rt,x_min,x_max,y_min,y_max]=...
    Triangle(B,xxx_1,yyy_1,xxx_2,yyy_2,xxx_3,yyy_3)
%Input: image matrix B,
%the coordinates (xxx_i,yyy_i), i=1,2,3, of the
%vertices of a triangle;

%Output: strings of the boundaries of the triangle,
%the ranges for x- and y-coordinates of the points
%inside the triangle

[M,N]=size(B); x1=yyy_1; x2=yyy_2; x3=yyy_3; y1=M+1-xxx_1;
y2=M+1-xxx_2; y3=M+1-xxx_3;

x1=x1+(x1==0)-(x1==N+1); x2=x2+(x2==0)-(x2==N+1);
x3=x3+(x3==0)-(x3==N+1);
```

```
y1=y1+(y1==0)-(y1==M+1); y2=y2+(y2==0)-(y2==M+1);
y3=y3+(y3==0)-(y3==M+1);

x_min=min([x1,x2,x3]); x_max=max([x1,x2,x3]); y_min=min([y1,y2,y3]);
y_max=max([y1,y2,y3]);

a=[x1,x2,x3,x1];b=[y1,y2,y3,y1]; if (x_min==x_max)
    low(x_min)=y_min;upp(x_max)=y_max;
    for j=y_min:y_max
        lt(j)=x_min;rt(j)=x_min;
    end
elseif (y_min==y_max)
    lt(y_min)=x_min;rt(y_min)=x_max;
    for i=x_min:x_max
        low(i)=y_min;upp(i)=y_min;
    end
else
    for i=x_min:x_max
      c=[i,i];d=[y_min,y_max];
      [p1,p2]=polyxpoly(a,b,c,d);
      low(i)=round(min(p2));upp(i)=round(max(p2));
    end
    for j=y_min:y_max
      e=[x_min,x_max];f=[j,j];
      [p3,p4]=polyxpoly(a,b,e,f);
      lt(j)=round(min(p3));rt(j)=round(max(p3));
    end
end
```

Centroid of the black area

```
%Centroid of the 'black' area
```

```
function [x_m,y_m]=Center_Tr(B,x_1,y_1,x_2,y_2,x_3,y_3)
%Input: image matrix B; (x_i,y_i) -- coordinates of
%the vertices of a triangle (i=1,2,3);

%Output: (x_m,y_m) -- coordinates of the centroid of
%the 'black' area inside the triangle;
[M,N]=size(B);

[low,upp,lt,rt,x_min,x_max,y_min,y_max]=...
    Triangle(B,x_1,y_1,x_2,y_2,x_3,y_3);

Area=0; Mx=0;My=0;

for i=x_min:x_max
 for j=low(i):upp(i)
    if (B(M+1-j,i)<0.8)
       Area=Area+1;
       My=My+i;
       Mx=Mx+j;
    end
 end
end

if (Area==0)
     x_m=0;
     y_m=0;
else
    y_m=round(My/Area);
    x_m=round(M+1-Mx/Area);
end
```

Centroid of the white area

```
%Centroid of the 'white' area
function [x_m,y_m]=Center_Trwhite(B,x_1,y_1,x_2,y_2,x_3,y_3)
%Input: image matrix B; (x_i,y_i) -- coordinates of
%the vertices of a triangle (i=1,2,3);

%Output: coordinates (x_m,y_m) of the centroid of
%the 'white' area inside the triangle;

[M,N]=size(B);

[low,upp,lt,rt,x_min,x_max,y_min,y_max]=...
    Triangle(B,x_1,y_1,x_2,y_2,x_3,y_3);

Area=0; Mx=0; My=0;

for i=x_min:x_max
  for j=low(i):upp(i)
    if (B(M+1-j,i)>=0.8)
     Area=Area+1;
     My=My+i;
     Mx=Mx+j;
    end
  end
end

if (Area==0)
    x_m=round((x_1+x_2+x_3)/3);
    y_m=round((y_1+y_2+y_3)/3);
else y_m=round(My/Area);
```

```
     x_m=round(M+1-Mx/Area);
end
```

Initial partitioning of an image

```
%Initial partitioning of an image
function [x_m, y_m]=Center_In(B)
%Input: image matrix B;

%Output: (x_m,y_m) -- coordinates of the
%centroid of the 'black' area in the image;

[M,N]=size(B);

Area=0;
 My=0;
 Mx=0;

for i=1:M
 for j=1:N
   if B(i,j)<0.8
    Area=Area+1;
    My=My+j;
    Mx=Mx+M-i+1;
   end
 end
end
 y_m=round(My/Area);
 x_m=round(M+1-Mx/Area);
```

Error over triangle

```
%Error over triangle
function [mean,eps]=Error(B,xx_1,yy_1,xx_2,yy_2,xx_3,yy_3)
%Input: image matrix B; coordinates of the vertices
%(xx_i,yy_i), i=1,2,3, of a triangle;

%Output: mean -- approximant of the image over triangle
%(0 or 1); the error of the approximation by 0 or 1;

[M,N]=size(B);

[low,upp,lt,rt,x_min,x_max,y_min,y_max]=...
    Triangle(B,xx_1,yy_1,xx_2,yy_2,xx_3,yy_3);

Lone=0;Area=0;

for i=x_min:x_max
  for j=low(i):upp(i)
   Lone=Lone+B(M+1-j,i);
   Area=Area+1;
  end
end

%Area -- area of the triangle
%Lone -- L_1-norm of the characteristic function
%over the triangle;

if (2*Lone>Area)
    mean=1;eps=Area-Lone;
else
    mean=0;eps=Lone;
```

```
end
```

Draw a line between two pixels

```
%Draw the straight line segment between two pixels;
function[G]=Draw(B,D,x_1,y_1,x_2,y_2)

%Input: image matrix B;
%existing triangulation on B;
%coordinates of two points
%(x_i,y_i), i=1,2, to be joined;

%Output: adds red straight line segment
%to the image joining two points;

[M,N]=size(B);

[low,upp,lt,rt,x_min,x_max,y_min,y_max]=...
    Triangle(B,x_1,y_1,x_2,y_2,x_2,y_2);
G=D;

for i=x_min:x_max
 for j=y_min:y_max
   if (i==lt(j))||(i==rt(j))||(j==low(i))||(j==upp(i))
    G(M+1-j,i,1)=1;
    G(M+1-j,i,2)=0;
    G(M+1-j,i,3)=0;
   end
 end
end
```

Main Parts of the Code

Subdivision procedure

```
%Implementation of the Main Subdivision Algorithm
function [D,Im,number]=...
    Division(B,D,Im,xx1,yy1,xx2,yy2,xx3,yy3,pres)

%Input: image matrix B;
%D - image with intriangulation;
%Im - (old) approximation of the image
%(xxi,yyi) -- coordinates of the vertices of
             %a triangle (i=1,2,3);
%pres - local error maximum

%Output: D - image with final triangulation
%Im - (new) approximation of the image;
%number - number of triangles in
%         the obtained triangulation


clear V

[M,N]=size(B);

V(1,1,1)=xx1+(xx1==0)-(xx1==M+1); V(1,1,2)=yy1+(yy1==0)-(yy1==N+1);
V(1,1,3)=xx2+(xx2==0)-(xx2==M+1); V(1,1,4)=yy2+(yy2==0)-(yy2==N+1);
V(1,1,5)=xx3+(xx3==0)-(xx3==M+1); V(1,1,6)=yy3+(yy3==0)-(yy3==N+1);

D=Draw(B,D,V(1,1,1),V(1,1,2),V(1,1,3),V(1,1,4));
D=Draw(B,D,V(1,1,1),V(1,1,2),V(1,1,5),V(1,1,6));
D=Draw(B,D,V(1,1,3),V(1,1,4),V(1,1,5),V(1,1,6)); [mean,eps]=...
Error(B,V(1,1,1),V(1,1,2),V(1,1,3),V(1,1,4),V(1,1,5),V(1,1,6));
```

```
V(1,1,7)=eps; k=1;level=1;num=0;

while (eps>pres)

    num=0;level=level+1;

for i=1:k

   x_1=V(level-1,i,1);y_1=V(level-1,i,2);
   x_2=V(level-1,i,3);y_2=V(level-1,i,4);
   x_3=V(level-1,i,5);y_3=V(level-1,i,6);
   err=V(level-1,i,7);

  if (err<=pres)
    V(level,num+1,1)=x_1;V(level,num+1,2)=y_1;
    V(level,num+1,3)=x_2;V(level,num+1,4)=y_2;
    V(level,num+1,5)=x_3;V(level,num+1,6)=y_3;
    [mean,er]=Error(B,x_1,y_1,x_2,y_2,x_3,y_3);
    V(level,num+1,7)=er;
   num=num+1;

  else

   K=[x_2-x_1,x_3-x_1;y_2-y_1,y_3-y_1];

    if (det(K)==0)

      [x_m1,y_m1]=Center_Tr(B,x_1,y_1,x_2,y_2,x_3,y_3);
      [x_m2,y_m2]=Center_Trwhite(B,x_1,y_1,x_2,y_2,x_3,y_3);
      x_m=round((x_m1+x_m2)/2);y_m=round((y_m1+y_m2)/2);
```

```
  x_11=min([x_1,x_2,x_3]);y_11=min([y_1,y_2,y_3]);
  x_22=max([x_1,x_2,x_3]);y_22=max([y_1,y_2,y_3]);

if (det([x_11-x_22,y_11-y_22;x_2-x_1,y_2-y_1])==0)&&...
        (det([x_11-x_22,y_11-y_22;x_3-x_1,y_3-y_1])==0)

  V(level,num+1,1)=x_11;V(level,num+1,2)=y_11;
  V(level,num+1,3)=x_m;V(level,num+1,4)=y_m;
  V(level,num+1,5)=x_m;V(level,num+1,6)=y_m;
  [mean,er]=Error(B,x_11,y_11,x_m,y_m,x_m,y_m);
  V(level,num+1,7)=er;

  V(level,num+2,1)=x_m;V(level,num+2,2)=y_m;
  V(level,num+2,3)=x_m;V(level,num+2,4)=y_m;
  V(level,num+2,5)=x_22;V(level,num+2,6)=y_22;
  [mean,er]=Error(B,x_m,y_m,x_m,y_m,x_22,y_22);
  V(level,num+2,7)=er;

else

  V(level,num+1,1)=x_22;V(level,num+1,2)=y_11;
  V(level,num+1,3)=x_m;V(level,num+1,4)=y_m;
  V(level,num+1,5)=x_m;V(level,num+1,6)=y_m;
  [mean,er]=Error(B,x_22,y_11,x_m,y_m,x_m,y_m);
  V(level,num+1,7)=er;

  V(level,num+2,1)=x_m;V(level,num+2,2)=y_m;
  V(level,num+2,3)=x_m;V(level,num+2,4)=y_m;
  V(level,num+2,5)=x_11;V(level,num+2,6)=y_22;
```

```
  [mean,er]=Error(B,x_m,y_m,x_m,y_m,x_11,y_22);
  V(level,num+2,7)=er;

end

num=num+2;

else

color=...
[(B(x_1,y_1)<0.8),(B(x_2,y_2)<0.8),(B(x_3,y_3)<0.8)];
% 1 if black 0 if white
colorsum=sum(color);

  if (colorsum==3)

    x_m=round((x_1+x_2+x_3)/3);
    y_m=round((y_1+y_2+y_3)/3);

    [D]=Draw(B,D,x_m,y_m,x_1,y_1);
    [D]=Draw(B,D,x_m,y_m,x_2,y_2);
    [D]=Draw(B,D,x_m,y_m,x_3,y_3);

    V(level,num+1,1)=x_1;V(level,num+1,2)=y_1;
    V(level,num+1,3)=x_2;V(level,num+1,4)=y_2;
    V(level,num+1,5)=x_m;V(level,num+1,6)=y_m;
    [mean,er]=Error(B,x_1,y_1,x_2,y_2,x_m,y_m);
    V(level,num+1,7)=er;

    V(level,num+2,1)=x_1;V(level,num+2,2)=y_1;
```

```
    V(level,num+2,3)=x_m;V(level,num+2,4)=y_m;
    V(level,num+2,5)=x_3;V(level,num+2,6)=y_3;
    [mean,er]=Error(B,x_1,y_1,x_m,y_m,x_3,y_3);
    V(level,num+2,7)=er;

    V(level,num+3,1)=x_m;V(level,num+3,2)=y_m;
    V(level,num+3,3)=x_2;V(level,num+3,4)=y_2;
    V(level,num+3,5)=x_3;V(level,num+3,6)=y_3;
    [mean,er]=Error(B,x_m,y_m,x_2,y_2,x_3,y_3);
    V(level,num+3,7)=er;

    num=num+3;
 end

if (colorsum==0)

    [x_m,y_m]=Center_Tr(B,x_1,y_1,x_2,y_2,x_3,y_3);

    [D]=Draw(B,D,x_m,y_m,x_1,y_1);
    [D]=Draw(B,D,x_m,y_m,x_2,y_2);
    [D]=Draw(B,D,x_m,y_m,x_3,y_3);

    V(level,num+1,1)=x_1;V(level,num+1,2)=y_1;
    V(level,num+1,3)=x_2;V(level,num+1,4)=y_2;
    V(level,num+1,5)=x_m;V(level,num+1,6)=y_m;
    [mean,er]=Error(B,x_1,y_1,x_2,y_2,x_m,y_m);
    V(level,num+1,7)=er;

    V(level,num+2,1)=x_1;V(level,num+2,2)=y_1;
    V(level,num+2,3)=x_m;V(level,num+2,4)=y_m;
```

```
        V(level,num+2,5)=x_3;V(level,num+2,6)=y_3;
        [mean,er]=Error(B,x_1,y_1,x_m,y_m,x_3,y_3);
        V(level,num+2,7)=er;

        V(level,num+3,1)=x_m;V(level,num+3,2)=y_m;
        V(level,num+3,3)=x_2;V(level,num+3,4)=y_2;
        V(level,num+3,5)=x_3;V(level,num+3,6)=y_3;
        [mean,er]=Error(B,x_m,y_m,x_2,y_2,x_3,y_3);
        V(level,num+3,7)=er;

        num=num+3;
    end

    if (colorsum==2)

%Case 1.

     if (color(1)==0)

        [x_m,y_m]=...
            Center_Trwhite(B,x_1,y_1,x_2,y_2,x_3,y_3);
        %pixels with intensity >0.8 = black

        E=[x_2-x_1,x_3-x_1;y_2-y_1,y_3-y_1];
        F=[3*x_m-3*x_1;3*y_m-3*y_1];
        cd=E\F;
        lambda=cd(1); mu=cd(2);
        c1=round(x_1+lambda*(x_2-x_1));
        d1=round(y_1+lambda*(y_2-y_1));
        c2=round(x_1+mu*(x_3-x_1));
```

```
d2=round(y_1+mu*(y_3-y_1));

x_mw=round((c1+c2)/2); y_mw=round((d1+d2)/2);

V(level,num+1,1)=c1;V(level,num+1,2)=d1;
V(level,num+1,3)=x_mw;V(level,num+1,4)=y_mw;
V(level,num+1,5)=x_1;V(level,num+1,6)=y_1;
[mean,er]=Error(B,x_1,y_1,c1,d1,x_mw,y_mw);
V(level,num+1,7)=er;

V(level,num+5,1)=c2;V(level,num+5,2)=d2;
V(level,num+5,3)=x_mw;V(level,num+5,4)=y_mw;
V(level,num+5,5)=x_1;V(level,num+5,6)=y_1;
[mean,er]=Error(B,x_1,y_1,c2,d2,x_mw,y_mw);
V(level,num+5,7)=er;

[D]=Draw(B,D,x_1,y_1,x_mw,y_mw);
[D]=Draw(B,D,c1,d1,c2,d2);
[D]=Draw(B,D,x_2,y_2,x_mw,y_mw);
[D]=Draw(B,D,x_3,y_3,x_mw,y_mw);

V(level,num+2,1)=c1;V(level,num+2,2)=d1;
V(level,num+2,3)=x_2;V(level,num+2,4)=y_2;
V(level,num+2,5)=x_mw;V(level,num+2,6)=y_mw;
[mean,er]=Error(B,x_mw,y_mw,c1,d1,x_2,y_2);
V(level,num+2,7)=er;

V(level,num+3,1)=c2;V(level,num+3,2)=d2;
V(level,num+3,3)=x_mw;V(level,num+3,4)=y_mw;
V(level,num+3,5)=x_3;V(level,num+3,6)=y_3;
```

```
        [mean,er]=Error(B,x_mw,y_mw,c2,d2,x_3,y_3);
        V(level,num+3,7)=er;

        V(level,num+4,1)=x_3;V(level,num+4,2)=y_3;
        V(level,num+4,3)=x_2;V(level,num+4,4)=y_2;
        V(level,num+4,5)=x_mw;V(level,num+4,6)=y_mw;
        [mean,er]=Error(B,x_2,y_2,x_mw,y_mw,x_3,y_3);
        V(level,num+4,7)=er;

        num=num+5;
    end

%Case 2.
    if (color(2)==0)
        [x_m,y_m]=...
            Center_Trwhite(B,x_1,y_1,x_2,y_2,x_3,y_3);

        %pixels with intensity >0.8 = black

        E=[x_1-x_2,x_3-x_2;y_1-y_2,y_3-y_2];
        F=[3*x_m-3*x_2;3*y_m-3*y_2];
        cd=E\F;
        lambda=cd(1); mu=cd(2);
        c1=round(x_2+lambda*(x_1-x_2));
        d1=round(y_2+lambda*(y_1-y_2));
        c2=round(x_2+mu*(x_3-x_2));
        d2=round(y_2+mu*(y_3-y_2));

        x_mw=round((c1+c2)/2); y_mw=round((d1+d2)/2);
```

```
V(level,num+1,1)=c1;V(level,num+1,2)=d1;
V(level,num+1,3)=x_mw;V(level,num+1,4)=y_mw;
V(level,num+1,5)=x_2;V(level,num+1,6)=y_2;
[mean,er]=Error(B,x_2,y_2,c1,d1,x_mw,y_mw);
V(level,num+1,7)=er;

V(level,num+5,1)=c2;V(level,num+5,2)=d2;
V(level,num+5,3)=x_mw;V(level,num+5,4)=y_mw;
V(level,num+5,5)=x_2;V(level,num+5,6)=y_2;
[mean,er]=Error(B,x_2,y_2,c2,d2,x_mw,y_mw);
V(level,num+5,7)=er;

[D]=Draw(B,D,x_2,y_2,x_mw,y_mw);
[D]=Draw(B,D,c1,d1,c2,d2);
[D]=Draw(B,D,x_1,y_1,x_mw,y_mw);
[D]=Draw(B,D,x_3,y_3,x_mw,y_mw);

V(level,num+2,1)=c1;V(level,num+2,2)=d1;
V(level,num+2,3)=x_1;V(level,num+2,4)=y_1;
V(level,num+2,5)=x_mw;V(level,num+2,6)=y_mw;
[mean,er]=Error(B,x_mw,y_mw,c1,d1,x_1,y_1);
V(level,num+2,7)=er;

V(level,num+3,1)=c2;V(level,num+3,2)=d2;
V(level,num+3,3)=x_mw;V(level,num+3,4)=y_mw;
V(level,num+3,5)=x_3;V(level,num+3,6)=y_3;
[mean,er]=Error(B,x_mw,y_mw,c2,d2,x_3,y_3);
V(level,num+3,7)=er;

V(level,num+4,1)=x_3;V(level,num+4,2)=y_3;
```

```
        V(level,num+4,3)=x_1;V(level,num+4,4)=y_1;
        V(level,num+4,5)=x_mw;V(level,num+4,6)=y_mw;
        [mean,er]=Error(B,x_1,y_1,x_mw,y_mw,x_3,y_3);
        V(level,num+4,7)=er;

        num=num+5;

    end

% Case 3.
    if (color(3)==0)
        [x_m,y_m]=...
            Center_Trwhite(B,x_1,y_1,x_2,y_2,x_3,y_3);
        %pixels with intensity >0.8 = black

        E=[x_2-x_3,x_1-x_3;y_2-y_3,y_1-y_3];
        F=[3*x_m-3*x_3;3*y_m-3*y_3];
        cd=E\F;
        lambda=cd(1); mu=cd(2);
        c1=round(x_3+lambda*(x_2-x_3));
        d1=round(y_3+lambda*(y_2-y_3));
        c2=round(x_3+mu*(x_1-x_3));
        d2=round(y_3+mu*(y_1-y_3));

        x_mw=round((c1+c2)/2); y_mw=round((d1+d2)/2);

        V(level,num+1,1)=c1;V(level,num+1,2)=d1;
        V(level,num+1,3)=x_mw;V(level,num+1,4)=y_mw;
        V(level,num+1,5)=x_3;V(level,num+1,6)=y_3;
        [mean,er]=Error(B,x_3,y_3,c1,d1,x_mw,y_mw);
```

```
V(level,num+1,7)=er;

V(level,num+5,1)=c2;V(level,num+5,2)=d2;
V(level,num+5,3)=x_mw;V(level,num+5,4)=y_mw;
V(level,num+5,5)=x_3;V(level,num+5,6)=y_3;
[mean,er]=Error(B,x_3,y_3,c2,d2,x_mw,y_mw);
V(level,num+5,7)=er;

[D]=Draw(B,D,x_1,y_1,x_mw,y_mw);
[D]=Draw(B,D,c1,d1,c2,d2);
[D]=Draw(B,D,x_2,y_2,x_mw,y_mw);
[D]=Draw(B,D,x_1,y_1,x_mw,y_mw);

V(level,num+2,1)=c1;V(level,num+2,2)=d1;
V(level,num+2,3)=x_2;V(level,num+2,4)=y_2;
V(level,num+2,5)=x_mw;V(level,num+2,6)=y_mw;
[mean,er]=Error(B,x_mw,y_mw,c1,d1,x_2,y_2);
V(level,num+2,7)=er;

V(level,num+3,1)=c2;V(level,num+3,2)=d2;
V(level,num+3,3)=x_mw;V(level,num+3,4)=y_mw;
V(level,num+3,5)=x_1;V(level,num+3,6)=y_1;
[mean,er]=Error(B,x_mw,y_mw,c2,d2,x_1,y_1);
V(level,num+3,7)=er;

V(level,num+4,1)=x_1;V(level,num+4,2)=y_1;
V(level,num+4,3)=x_2;V(level,num+4,4)=y_2;
V(level,num+4,5)=x_mw;V(level,num+4,6)=y_mw;
[mean,er]=Error(B,x_2,y_2,x_mw,y_mw,x_1,y_1);
V(level,num+4,7)=er;
```

```
        num=num+5;
    end
clear col clear low clear upp clear lt clear rt

   end

   if (colorsum==1)
%Case 1.

     if (color(1)==1)

        [x_mw,y_mw]=...
            Center_Tr(B,x_2,y_2,x_2,y_2,x_3,y_3);

      if (x_mw==0)

          [x_m,y_m]=Center_Tr(B,x_1,y_1,x_2,y_2,x_3,y_3);
          %pixels with intensity >0.8 = black

          E=[x_2-x_1,x_3-x_1;y_2-y_1,y_3-y_1];
          F=[3*x_m-3*x_1;3*y_m-3*y_1];
          cd=E\F;
          lambda=cd(1); mu=cd(2);
          c1=round(x_1+lambda*(x_2-x_1));
          d1=round(y_1+lambda*(y_2-y_1));
          c2=round(x_1+mu*(x_3-x_1));
          d2=round(y_1+mu*(y_3-y_1));

        if (c1<min(x_1,x_2))||(c1>max(x_1,x_2))
```

```
a1=[x_2,x_3]; b1=[y_2,y_3];
a2=[c1,c2];b2=[d1,d2];
[e1,f1]=polyxpoly(a1,b1,a2,b2);
x_med=round((c1+c2)/2);y_med=round((d1+d2)/2);
c1=round(e1);d1=round(f1);

V(level,num+1,1)=c1;V(level,num+1,2)=d1;
V(level,num+1,3)=x_2;V(level,num+1,4)=y_2;
V(level,num+1,5)=x_1;V(level,num+1,6)=y_1;
[mean,er]=Error(B,x_1,y_1,c1,d1,x_2,y_2);
V(level,num+1,7)=er;

V(level,num+2,1)=c1;V(level,num+2,2)=d1;
V(level,num+2,3)=x_med;V(level,num+2,4)=y_med;
V(level,num+2,5)=x_1;V(level,num+2,6)=y_1;
[mean,er]=Error(B,x_1,y_1,c1,d1,x_med,y_med);
V(level,num+2,7)=er;

V(level,num+3,1)=c1;V(level,num+3,2)=d1;
V(level,num+3,3)=x_med;V(level,num+3,4)=y_med;
V(level,num+3,5)=x_3;V(level,num+3,6)=y_3;
[mean,er]=Error(B,x_3,y_3,c1,d1,x_med,y_med);
V(level,num+3,7)=er;

V(level,num+4,1)=x_med;V(level,num+4,2)=y_med;
V(level,num+4,3)=c2;V(level,num+4,4)=d2;
V(level,num+4,5)=x_1;V(level,num+4,6)=y_1;
[mean,er]=Error(B,x_1,y_1,x_med,y_med,c2,d2);
V(level,num+4,7)=er;
```

```
    V(level,num+5,1)=x_med;V(level,num+5,2)=y_med;
    V(level,num+5,3)=c2;V(level,num+5,4)=d2;
    V(level,num+5,5)=x_3;V(level,num+5,6)=y_3;
    [mean,er]=Error(B,x_3,y_3,x_med,y_med,c2,d2);
    V(level,num+5,7)=er;

    [D]=Draw(B,D,c1,d1,c2,d2);
    [D]=Draw(B,D,c1,d1,x_1,y_1);
    [D]=Draw(B,D,x_med,y_med,x_3,y_3);
    [D]=Draw(B,D,x_med,y_med,x_1,y_1);

    num=num+5;

elseif (c2<min(x_1,x_3))||(c2>max(x_1,x_3))
    a1=[x_2,x_3]; b1=[y_2,y_3];
    a2=[c1,c2];b2=[d1,d2];
    [e1,f1]=polyxpoly(a1,b1,a2,b2);
    x_med=round((c1+c2)/2);y_med=round((d1+d2)/2);
    c2=round(e1);d2=round(f1);

    V(level,num+1,1)=c2;V(level,num+1,2)=d2;
    V(level,num+1,3)=x_3;V(level,num+1,4)=y_3;
    V(level,num+1,5)=x_1;V(level,num+1,6)=y_1;
    [mean,er]=Error(B,x_1,y_1,c2,d2,x_3,y_3);
    V(level,num+1,7)=er;

    V(level,num+2,1)=c2;V(level,num+2,2)=d2;
    V(level,num+2,3)=x_med;V(level,num+2,4)=y_med;
    V(level,num+2,5)=x_1;V(level,num+2,6)=y_1;
    [mean,er]=Error(B,x_1,y_1,c2,d2,x_med,y_med);
```

```
        V(level,num+2,7)=er;

        V(level,num+3,1)=c2;V(level,num+3,2)=d2;
        V(level,num+3,3)=x_med;V(level,num+3,4)=y_med;
        V(level,num+3,5)=x_2;V(level,num+3,6)=y_2;
        [mean,er]=Error(B,x_2,y_2,c2,d2,x_med,y_med);
        V(level,num+3,7)=er;

        V(level,num+4,1)=x_med;V(level,num+4,2)=y_med;
        V(level,num+4,3)=c1;V(level,num+4,4)=d1;
        V(level,num+4,5)=x_1;V(level,num+4,6)=y_1;
        [mean,er]=Error(B,x_1,y_1,x_med,y_med,c1,d1);
        V(level,num+4,7)=er;

        V(level,num+5,1)=x_med;V(level,num+5,2)=y_med;
        V(level,num+5,3)=c1;V(level,num+5,4)=d1;
        V(level,num+5,5)=x_2;V(level,num+5,6)=y_2;
        [mean,er]=Error(B,x_2,y_2,x_med,y_med,c1,d1);
        V(level,num+5,7)=er;

        [D]=Draw(B,D,c1,d1,c2,d2);
        [D]=Draw(B,D,c2,d2,x_1,y_1);
        [D]=Draw(B,D,x_med,y_med,x_2,y_2);
        [D]=Draw(B,D,x_med,y_med,x_1,y_1);

        num=num+5;

    else

       x_med=round((c1+c2)/2);y_med=round((d1+d2)/2);
```

```
c1=c1+(c1==0)-(c1==M+1);c2=c2+(c2==0)-(c2==M+1);
d1=d1+(d1==0)-(d1==N+1);
d2=d2+(d2==0)-(d2==N+1);

V(level,num+1,1)=c1;V(level,num+1,2)=d1;
V(level,num+1,3)=x_med;V(level,num+1,4)=y_med;
V(level,num+1,5)=x_1;V(level,num+1,6)=y_1;
[mean,er]=Error(B,x_1,y_1,c1,d1,x_med,y_med);
V(level,num+1,7)=er;

V(level,num+2,1)=c1;V(level,num+2,2)=d1;
V(level,num+2,3)=x_med;V(level,num+2,4)=y_med;
V(level,num+2,5)=x_2;V(level,num+2,6)=y_2;
[mean,er]=Error(B,x_med,y_med,c1,d1,x_2,y_2);
V(level,num+2,7)=er;

V(level,num+3,1)=x_2;V(level,num+3,2)=y_2;
V(level,num+3,3)=x_med;V(level,num+3,4)=y_med;
V(level,num+3,5)=x_3;V(level,num+3,6)=y_3;
[mean,er]=Error(B,x_med,y_med,x_2,y_2,x_3,y_3);
V(level,num+3,7)=er;

V(level,num+4,1)=c2;V(level,num+4,2)=d2;
V(level,num+4,3)=x_3;V(level,num+4,4)=y_3;
V(level,num+4,5)=x_med;V(level,num+4,6)=y_med;
[mean,er]=Error(B,x_3,y_3,x_med,y_med,c2,d2);
V(level,num+4,7)=er;

V(level,num+5,1)=x_med;V(level,num+5,2)=y_med;
```

```
        V(level,num+5,3)=c2;V(level,num+5,4)=d2;
        V(level,num+5,5)=x_1;V(level,num+5,6)=y_1;
        [mean,er]=Error(B,x_1,y_1,x_med,y_med,c2,d2);
        V(level,num+5,7)=er;

        [D]=Draw(B,D,c1,d1,c2,d2);
        [D]=Draw(B,D,x_med,y_med,x_1,y_1);
        [D]=Draw(B,D,x_med,y_med,x_2,y_2);
        [D]=Draw(B,D,x_med,y_med,x_3,y_3);

        num=num+5;
    end

else

    x_m=x_mw;y_m=y_mw;

    [D]=Draw(B,D,x_m,y_m,x_1,y_1);

    x_m=x_m+(x_m==0)-(x_m==M+1);
    y_m=y_m+(y_m==0)-(y_m==M+1);


    V(level,num+1,1)=x_1;V(level,num+1,2)=y_1;
    V(level,num+1,3)=x_2;V(level,num+1,4)=y_2;
    V(level,num+1,5)=x_m;V(level,num+1,6)=y_m;
    [mean,er]=Error(B,x_1,y_1,x_2,y_2,x_m,y_m);
    V(level,num+1,7)=er;

    V(level,num+2,1)=x_1;V(level,num+2,2)=y_1;
```

```
          V(level,num+2,3)=x_3;V(level,num+2,4)=y_3;
          V(level,num+2,5)=x_m;V(level,num+2,6)=y_m;
          [mean,er]=Error(B,x_1,y_1,x_m,y_m,x_3,y_3);
          V(level,num+2,7)=er;

          num=num+2;

      end
clear col
    clear low
    clear upp
    clear lt
    clear rt
     end


%Case 2.
if (color(2)==1)

    [x_mw,y_mw]=Center_Tr(B,x_1,y_1,x_1,y_1,x_3,y_3);
    if (x_mw==0)

        [x_m,y_m]=...
            Center_Tr(B,x_1,y_1,x_2,y_2,x_3,y_3);

        E=[x_1-x_2,x_3-x_2;y_1-y_2,y_3-y_2];
        F=[3*x_m-3*x_2;3*y_m-3*y_2];
        cd=E\F;
        lambda=cd(1); mu=cd(2);
        c1=round(x_2+lambda*(x_1-x_2));
```

```
d1=round(y_2+lambda*(y_1-y_2));
c2=round(x_2+mu*(x_3-x_2));
d2=round(y_2+mu*(y_3-y_2));

if (c1<min(x_2,x_1))||(c1>max(x_1,x_2))
    a1=[x_1,x_3]; b1=[y_1,y_3];
    a2=[c1,c2];b2=[d1,d2];
    [e1,f1]=polyxpoly(a1,b1,a2,b2);
    x_med=round((c1+c2)/2);y_med=round((d1+d2)/2);
    c1=round(e1);d1=round(f1);

    V(level,num+1,1)=c1;V(level,num+1,2)=d1;
    V(level,num+1,3)=x_1;V(level,num+1,4)=y_1;
    V(level,num+1,5)=x_2;V(level,num+1,6)=y_2;
    [mean,er]=Error(B,x_1,y_1,c1,d1,x_2,y_2);
    V(level,num+1,7)=er;

    V(level,num+2,1)=c1;V(level,num+2,2)=d1;
    V(level,num+2,3)=x_med;V(level,num+2,4)=y_med;
    V(level,num+2,5)=x_2;V(level,num+2,6)=y_2;
    [mean,er]=Error(B,x_2,y_2,c1,d1,x_med,y_med);
    V(level,num+2,7)=er;

    V(level,num+3,1)=c1;V(level,num+3,2)=d1;
    V(level,num+3,3)=x_med;V(level,num+3,4)=y_med;
    V(level,num+3,5)=x_3;V(level,num+3,6)=y_3;
    [mean,er]=Error(B,x_3,y_3,c1,d1,x_med,y_med);
    V(level,num+3,7)=er;

    V(level,num+4,1)=x_med;V(level,num+4,2)=y_med;
```

```
        V(level,num+4,3)=c2;V(level,num+4,4)=d2;
        V(level,num+4,5)=x_2;V(level,num+4,6)=y_2;
        [mean,er]=Error(B,x_2,y_2,x_med,y_med,c2,d2);
        V(level,num+4,7)=er;

        V(level,num+5,1)=x_med;V(level,num+5,2)=y_med;
        V(level,num+5,3)=c2;V(level,num+5,4)=d2;
        V(level,num+5,5)=x_3;V(level,num+5,6)=y_3;
        [mean,er]=Error(B,x_3,y_3,x_med,y_med,c2,d2);
        V(level,num+5,7)=er;

        [D]=Draw(B,D,c1,d1,c2,d2);
        [D]=Draw(B,D,c1,d1,x_2,y_2);
        [D]=Draw(B,D,x_med,y_med,x_3,y_3);
        [D]=Draw(B,D,x_med,y_med,x_2,y_2);

        num=num+5;

    elseif (c2<min(x_3,x_2))||(c2>max(x_3,x_2))
        a1=[x_1,x_3]; b1=[y_1,y_3];
        a2=[c1,c2];b2=[d1,d2];
        [e1,f1]=polyxpoly(a1,b1,a2,b2);
        x_med=round((c1+c2)/2);y_med=round((d1+d2)/2);
        c2=round(e1);d2=round(f1);

        V(level,num+1,1)=c2;V(level,num+1,2)=d2;
        V(level,num+1,3)=x_3;V(level,num+1,4)=y_3;
        V(level,num+1,5)=x_2;V(level,num+1,6)=y_2;
        [mean,er]=Error(B,x_2,y_2,c2,d2,x_3,y_3);
        V(level,num+1,7)=er;
```

```
V(level,num+2,1)=c2;V(level,num+2,2)=d2;
V(level,num+2,3)=x_med;V(level,num+2,4)=y_med;
V(level,num+2,5)=x_2;V(level,num+2,6)=y_2;
[mean,er]=Error(B,x_2,y_2,c2,d2,x_med,y_med);
V(level,num+2,7)=er;

V(level,num+3,1)=c2;V(level,num+3,2)=d2;
V(level,num+3,3)=x_med;V(level,num+3,4)=y_med;
V(level,num+3,5)=x_1;V(level,num+3,6)=y_1;
[mean,er]=Error(B,x_1,y_1,c2,d2,x_med,y_med);
V(level,num+3,7)=er;

V(level,num+4,1)=x_med;V(level,num+4,2)=y_med;
V(level,num+4,3)=c1;V(level,num+4,4)=d1;
V(level,num+4,5)=x_2;V(level,num+4,6)=y_2;
[mean,er]=Error(B,x_2,y_2,x_med,y_med,c1,d1);
V(level,num+4,7)=er;

V(level,num+5,1)=x_med;V(level,num+5,2)=y_med;
V(level,num+5,3)=c1;V(level,num+5,4)=d1;
V(level,num+5,5)=x_1;V(level,num+5,6)=y_1;
[mean,er]=Error(B,x_1,y_1,x_med,y_med,c1,d1);
V(level,num+5,7)=er;

[D]=Draw(B,D,c1,d1,c2,d2);
[D]=Draw(B,D,c2,d2,x_2,y_2);
[D]=Draw(B,D,x_med,y_med,x_1,y_1);
[D]=Draw(B,D,x_med,y_med,x_2,y_2);
```

```
        num=num+5;

    else

        x_med=round((c1+c2)/2);y_med=round((d1+d2)/2);

        V(level,num+1,1)=c1;V(level,num+1,2)=d1;
        V(level,num+1,3)=x_med;V(level,num+1,4)=y_med;
        V(level,num+1,5)=x_2;V(level,num+1,6)=y_2;
        [mean,er]=Error(B,x_2,y_2,c1,d1,x_med,y_med);
        V(level,num+1,7)=er;

        V(level,num+2,1)=c1;V(level,num+2,2)=d1;
        V(level,num+2,3)=x_med;V(level,num+2,4)=y_med;
        V(level,num+2,5)=x_1;V(level,num+2,6)=y_1;
        [mean,er]=Error(B,x_med,y_med,c1,d1,x_1,y_1);
        V(level,num+2,7)=er;

        V(level,num+3,1)=x_1;V(level,num+3,2)=y_1;
        V(level,num+3,3)=x_med;V(level,num+3,4)=y_med;
        V(level,num+3,5)=x_3;V(level,num+3,6)=y_3;
        [mean,er]=Error(B,x_med,y_med,x_1,y_1,x_3,y_3);
        V(level,num+3,7)=er;

        V(level,num+4,1)=c2;V(level,num+4,2)=d2;
        V(level,num+4,3)=x_3;V(level,num+4,4)=y_3;
        V(level,num+4,5)=x_med;V(level,num+4,6)=y_med;
        [mean,er]=Error(B,x_3,y_3,x_med,y_med,c2,d2);
        V(level,num+4,7)=er;
```

```
            V(level,num+5,1)=x_med;V(level,num+5,2)=y_med;
            V(level,num+5,3)=c2;V(level,num+5,4)=d2;
            V(level,num+5,5)=x_2;V(level,num+5,6)=y_2;
            [mean,er]=Error(B,x_2,y_2,x_med,y_med,c2,d2);
            V(level,num+5,7)=er;

            [D]=Draw(B,D,c1,d1,c2,d2);
            [D]=Draw(B,D,x_med,y_med,x_1,y_1);
            [D]=Draw(B,D,x_med,y_med,x_2,y_2);
            [D]=Draw(B,D,x_med,y_med,x_3,y_3);

            num=num+5;

        end

    else

        x_m=x_mw;y_m=y_mw;

        [D]=Draw(B,D,x_m,y_m,x_2,y_2);

        V(level,num+1,1)=x_1;V(level,num+1,2)=y_1;
        V(level,num+1,3)=x_2;V(level,num+1,4)=y_2;
        V(level,num+1,5)=x_m;V(level,num+1,6)=y_m;
        [mean,er]=Error(B,x_1,y_1,x_2,y_2,x_m,y_m);
        V(level,num+1,7)=er;

        V(level,num+2,1)=x_2;V(level,num+2,2)=y_2;
        V(level,num+2,3)=x_3;V(level,num+2,4)=y_3;
        V(level,num+2,5)=x_m;V(level,num+2,6)=y_m;
```

```
        [mean,er]=Error(B,x_m,y_m,x_2,y_2,x_3,y_3);
        V(level,num+2,7)=er;

        num=num+2;

    end
    clear col
    clear low
    clear upp
    clear lt
    clear rt
end


%Case 3.
if (color(3)==1)

    [x_mw,y_mw]=Center_Tr(B,x_1,y_1,x_2,y_2,x_1,y_1);

    if (x_mw==0)

        [x_m,y_m]=Center_Tr(B,x_1,y_1,x_2,y_2,x_3,y_3);

        E=[x_2-x_3,x_1-x_3;y_2-y_3,y_1-y_3];
        F=[3*x_m-3*x_3;3*y_m-3*y_3];
        cd=E\F;
        lambda=cd(1); mu=cd(2);
        c1=round(x_3+lambda*(x_2-x_3));
        d1=round(y_3+lambda*(y_2-y_3));
        c2=round(x_3+mu*(x_1-x_3));
```

```
d2=round(y_3+mu*(y_1-y_3));

if (c1<min(x_3,x_2))||(c1>max(x_3,x_2))

    a1=[x_2,x_1]; b1=[y_2,y_1];
    a2=[c1,c2];b2=[d1,d2];
    [e1,f1]=polyxpoly(a1,b1,a2,b2);
    x_med=round((c1+c2)/2);y_med=round((d1+d2)/2);
    c1=round(e1);d1=round(f1);

    V(level,num+1,1)=c1;V(level,num+1,2)=d1;
    V(level,num+1,3)=x_2;V(level,num+1,4)=y_2;
    V(level,num+1,5)=x_3;V(level,num+1,6)=y_3;
    [mean,er]=Error(B,x_3,y_3,c1,d1,x_2,y_2);
    V(level,num+1,7)=er;

    V(level,num+2,1)=c1;V(level,num+2,2)=d1;
    V(level,num+2,3)=x_med;V(level,num+2,4)=y_med;
    V(level,num+2,5)=x_3;V(level,num+2,6)=y_3;
    [mean,er]=Error(B,x_3,y_3,c1,d1,x_med,y_med);
    V(level,num+2,7)=er;

    V(level,num+3,1)=c1;V(level,num+3,2)=d1;
    V(level,num+3,3)=x_med;V(level,num+3,4)=y_med;
    V(level,num+3,5)=x_1;V(level,num+3,6)=y_1;
    [mean,er]=Error(B,x_1,y_1,c1,d1,x_med,y_med);
    V(level,num+3,7)=er;

    V(level,num+4,1)=x_med;V(level,num+4,2)=y_med;
    V(level,num+4,3)=c2;V(level,num+4,4)=d2;
```

```
        V(level,num+4,5)=x_3;V(level,num+4,6)=y_3;
        [mean,er]=Error(B,x_3,y_3,x_med,y_med,c2,d2);
        V(level,num+4,7)=er;

        V(level,num+5,1)=x_med;V(level,num+5,2)=y_med;
        V(level,num+5,3)=c2;V(level,num+5,4)=d2;
        V(level,num+5,5)=x_1;V(level,num+5,6)=y_1;
        [mean,er]=Error(B,x_1,y_1,x_med,y_med,c2,d2);
        V(level,num+5,7)=er;

        [D]=Draw(B,D,c1,d1,c2,d2);
        [D]=Draw(B,D,c1,d1,x_1,y_1);
        [D]=Draw(B,D,x_med,y_med,x_3,y_3);
        [D]=Draw(B,D,x_med,y_med,x_1,y_1);

        num=num+5;

    elseif (c2<min(x_1,x_3))||(c2>max(x_1,x_3))

        a1=[x_2,x_1]; b1=[y_2,y_1];
        a2=[c1,c2];b2=[d1,d2];
        [e1,f1]=polyxpoly(a1,b1,a2,b2);
        x_med=round((c1+c2)/2);y_med=round((d1+d2)/2);
        c2=round(e1);d2=round(f1);

        V(level,num+1,1)=c2;V(level,num+1,2)=d2;
        V(level,num+1,3)=x_1;V(level,num+1,4)=y_1;
        V(level,num+1,5)=x_3;V(level,num+1,6)=y_3;
        [mean,er]=Error(B,x_1,y_1,c2,d2,x_3,y_3);
        V(level,num+1,7)=er;
```

```
V(level,num+2,1)=c2;V(level,num+2,2)=d2;
V(level,num+2,3)=x_med;V(level,num+2,4)=y_med;
V(level,num+2,5)=x_3;V(level,num+2,6)=y_3;
[mean,er]=Error(B,x_3,y_3,c2,d2,x_med,y_med);
V(level,num+2,7)=er;

V(level,num+3,1)=c2;V(level,num+3,2)=d2;
V(level,num+3,3)=x_med;V(level,num+3,4)=y_med;
V(level,num+3,5)=x_2;V(level,num+3,6)=y_2;
[mean,er]=Error(B,x_2,y_2,c2,d2,x_med,y_med);
V(level,num+3,7)=er;

V(level,num+4,1)=x_med;V(level,num+4,2)=y_med;
V(level,num+4,3)=c1;V(level,num+4,4)=d1;
V(level,num+4,5)=x_3;V(level,num+4,6)=y_3;
[mean,er]=Error(B,x_3,y_3,x_med,y_med,c1,d1);
V(level,num+4,7)=er;

V(level,num+5,1)=x_med;V(level,num+5,2)=y_med;
V(level,num+5,3)=c1;V(level,num+5,4)=d1;
V(level,num+5,5)=x_2;V(level,num+5,6)=y_2;
[mean,er]=Error(B,x_2,y_2,x_med,y_med,c1,d1);
V(level,num+5,7)=er;

[D]=Draw(B,D,c1,d1,c2,d2);
[D]=Draw(B,D,c2,d2,x_3,y_3);
[D]=Draw(B,D,x_med,y_med,x_2,y_2);
[D]=Draw(B,D,x_med,y_med,x_3,y_3);
```

```
        num=num+5;

    else

        x_med=round((c1+c2)/2);y_med=round((d1+d2)/2);

        V(level,num+1,1)=c1;V(level,num+1,2)=d1;
        V(level,num+1,3)=x_med;V(level,num+1,4)=y_med;
        V(level,num+1,5)=x_3;V(level,num+1,6)=y_3;
        [mean,er]=Error(B,x_3,y_3,c1,d1,x_med,y_med);
        V(level,num+1,7)=er;

        V(level,num+2,1)=c1;V(level,num+2,2)=d1;
        V(level,num+2,3)=x_med;V(level,num+2,4)=y_med;
        V(level,num+2,5)=x_2;V(level,num+2,6)=y_2;
        [mean,er]=Error(B,x_med,y_med,c1,d1,x_2,y_2);
        V(level,num+2,7)=er;

        V(level,num+3,1)=x_2;V(level,num+3,2)=y_2;
        V(level,num+3,3)=x_med;V(level,num+3,4)=y_med;
        V(level,num+3,5)=x_1;V(level,num+3,6)=y_1;
        [mean,er]=Error(B,x_med,y_med,x_2,y_2,x_1,y_1);
        V(level,num+3,7)=er;

        V(level,num+4,1)=c2;V(level,num+4,2)=d2;
        V(level,num+4,3)=x_1;V(level,num+4,4)=y_1;
        V(level,num+4,5)=x_med;V(level,num+4,6)=y_med;
        [mean,er]=Error(B,x_1,y_1,x_med,y_med,c2,d2);
        V(level,num+4,7)=er;
```

```
            V(level,num+5,1)=x_med;V(level,num+5,2)=y_med;
            V(level,num+5,3)=c2;V(level,num+5,4)=d2;
            V(level,num+5,5)=x_3;V(level,num+5,6)=y_3;
            [mean,er]=Error(B,x_3,y_3,x_med,y_med,c2,d2);
            V(level,num+5,7)=er;

            [D]=Draw(B,D,c1,d1,c2,d2);
            [D]=Draw(B,D,x_med,y_med,x_1,y_1);
            [D]=Draw(B,D,x_med,y_med,x_2,y_2);
            [D]=Draw(B,D,x_med,y_med,x_3,y_3);

            num=num+5;

        end

    else
        x_m=x_mw;y_m=y_mw;

        [D]=Draw(B,D,x_m,y_m,x_3,y_3);

        V(level,num+1,1)=x_1;V(level,num+1,2)=y_1;
        V(level,num+1,3)=x_3;V(level,num+1,4)=y_3;
        V(level,num+1,5)=x_m;V(level,num+1,6)=y_m;
        [mean,er]=Error(B,x_1,y_1,x_m,y_m,x_3,y_3);
        V(level,num+1,7)=er;

        V(level,num+2,1)=x_2;V(level,num+2,2)=y_2;
        V(level,num+2,3)=x_3;V(level,num+2,4)=y_3;
        V(level,num+2,5)=x_m;V(level,num+2,6)=y_m;
        [mean,er]=Error(B,x_m,y_m,x_2,y_2,x_3,y_3);
```

```
        V(level,num+2,7)=er;

        num=num+2;

    end
    clear col
    clear low
    clear upp
    clear lt
    clear rt
end
  end
   end
 end
end

eps=max(V(level,:,7));

k=num; end

number=k;

for m=1:k
    x_1=V(level,m,1);y_1=V(level,m,2);
    x_2=V(level,m,3);y_2=V(level,m,4);
    x_3=V(level,m,5);y_3=V(level,m,6);

    [mean,er]=Error(B,x_1,y_1,x_2,y_2,x_3,y_3);
    [low,upp,lt,rt,x_min,x_max,y_min,y_max]=...
        Triangle(B,x_1,y_1,x_2,y_2,x_3,y_3);
```

```
    for i=x_min:x_max
      for j=y_min:y_max
        if (i>=lt(j))&&(i<=rt(j))&&(j>=low(i))&&(j<=upp(i))
                Im(M+1-j,i)=mean;
        end
      end
    end
end
```

Input the image and output the result

```
function Triangulation(filename,pres)
%Input: image and presicion;

%Output: triangulated image and the approximation
%based on the triangulation

A=imread(filename); A=rgb2gray(A); A=mat2gray(A); [M,N]=size(A);

Im=A; Im=0; C(:,:,1)=A; C(:,:,2)=A; C(:,:,3)=A;



[x_m,y_m]=Center_In(A);
[C,Im,number1]=Division(A,C,Im,1,1,M,1,x_m,y_m,pres);
[C,Im,number2]=Division(A,C,Im,M,1,M,N,x_m,y_m,pres);
[C,Im,number3]=Division(A,C,Im,M,N,1,N,x_m,y_m,pres);
[C,Im,number4]=Division(A,C,Im,1,1,1,N,x_m,y_m,pres);

number1+number2+number3+number4

E=abs(A-Im); sum(sum(E))
```

```
figure(1); imshow(Im) figure(2); imshow(C)
```

Numerical Results

In the following comparison tables, we present the results of our adaptive approximation of the same image with three different local error precisions. To show the relation between the theoretical results established in Theorem 2.2 and the results of the corresponding implementations, we add two columns that show the *rates of decay* of the right-hand sides in the estimates established in Theorem 2.2:

Table A.1: Image of a square (Figure A.1a): analysis of adaptive approximation

Precision ϵ		*# of triangles*	*Global error*		$\#\mathcal{D}$	$\sigma_1(f,\mathcal{D})_1$
(in pixels)	*(in* cm^2*)*	*(final partition)*	*(in pixels)*	*(in* cm^2*)*	$\epsilon^{-1/3}\ln(1/\epsilon)$	$\epsilon^{2/3}\ln(1/\epsilon)$
200	0.14	40	1716	1.2	3.8	0.53
100	0.07	72	816	0.57	6.5	0.45
50	0.035	108	735	0.51	10	0.36

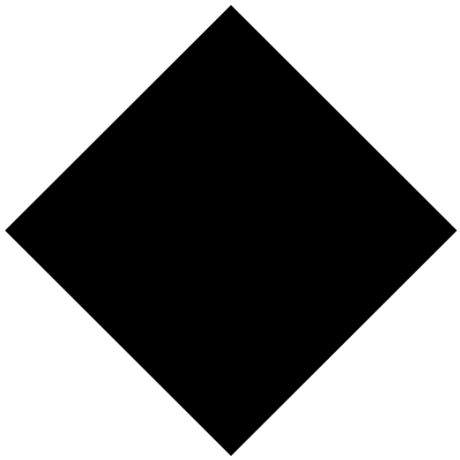
Table A.2: Image of a circle (Figure A.2a): analysis of adaptive approximation

Precision ϵ		*# of triangles*	*Global error*		$\#\mathcal{D}$	$\sigma_1(f,\mathcal{D})_1$
(in pixels)	*(in* cm^2*)*	*(final partition)*	*(in pixels)*	*(in* cm^2*)*	$\epsilon^{-1/3}\ln(1/\epsilon)$	$\epsilon^{2/3}\ln(1/\epsilon)$
400	0.28	92	2104	1.475	2	0.55
100	0.07	100	1397	0.98	6.5	0.45
50	0.035	140	1058	0.75	10	0.36

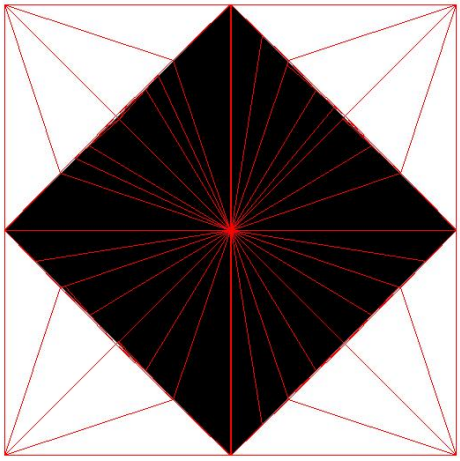
Table A.3: Image of a polygon (Figure A.3a): analysis of adaptive approximation

Precision ϵ		*# of triangles*	*Global error*		$\#\mathcal{D}$	$\sigma_1(f,\mathcal{D})_1$
(in pixels)	*(in* cm^2*)*	*(final partition)*	*(in pixels)*	*(in* cm^2*)*	$\epsilon^{-1/3}\ln(1/\epsilon)$	$\epsilon^{2/3}\ln(1/\epsilon)$
60	0.14	104	893	0.63	9	0.38
40	0.028	132	748	0.52	12	0.32
20	0.042	213	568	0.39	18	0.25

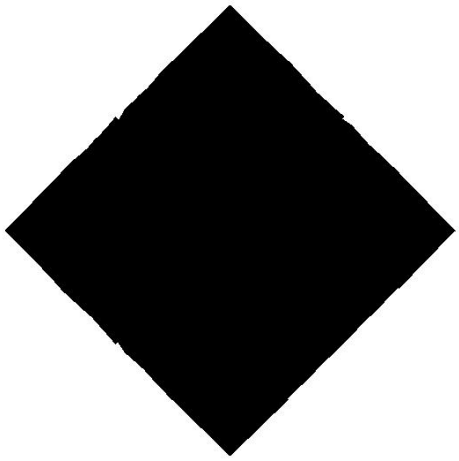
Figure A.1: **Implementation: square**



(a) 512×512 original image of a square

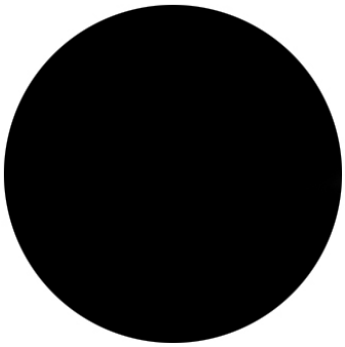


(b) Triangulation with 108 triangles

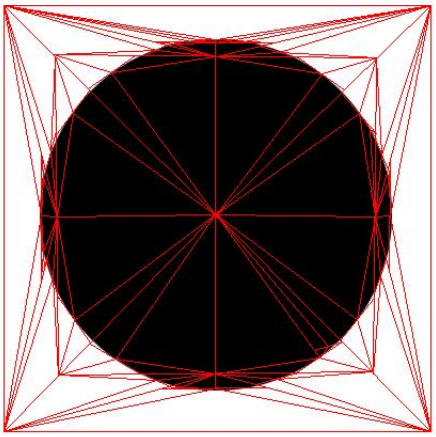


(c) Approximation of an image, local error = 50 pixels, total error= 735 pixels

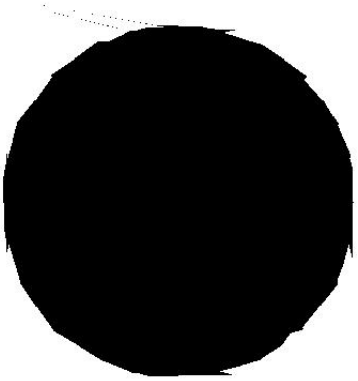
Figure A.2: **Implementation: circle**



(a) 400 × 400 original image of a circle

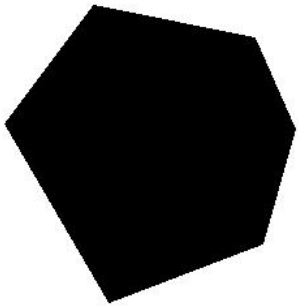


(b) Triangulation with 140 triangles

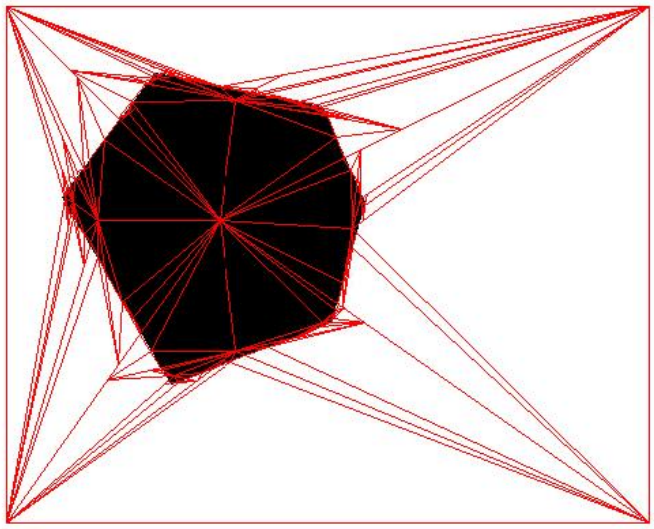


(c) Approximation of an image, local error = 50 pixels, total error= 1058 pixels

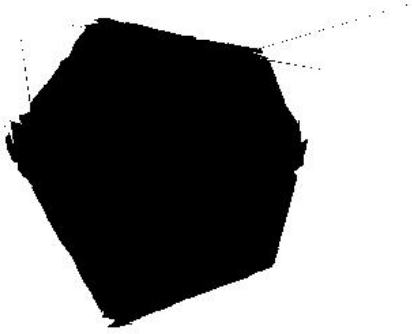
Figure A.3: **Implementation: polygon**



(a) 474 × 380 original image of a polygon



(b) Triangulation with 213 triangles



(c) Approximation of an image, local error = 20 pixels, total error= 568 pixels