The Convergence Rate of the Sandwich Algorithm for Approximating Convex Figures in the Plane

(extended abstract)

Günter Rote
University of Waterloo

The Sandwich algorithm is an iterative procedure for approximating a convex figure P in the plane by convex polygons.

For a set of n points ("knots") on the boundary of P, the convex polygon formed by these points is an inner approximation of P; a set of supporting lines at these points forms also a convex n-gon, which is an outer approximation of P (see figure 1). Since the body P that we are interested in lies between the inner and outer approximation, we call such an approximating pair of polygons a *Sandwich approximation* (cf. Burkard, Hamacher, and Rote [1990], Martelli [1962]).

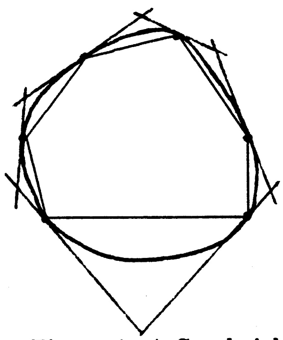

Figure 1: A Sandwich approximation with 6 knots

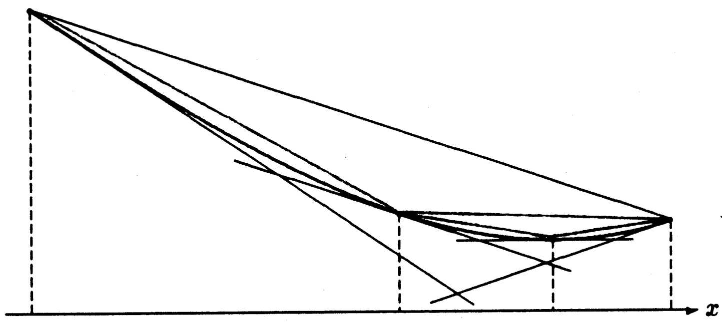

Figure 2: The Sandwich algorithm after two partitioning steps

The *Sandwich algorithm* starts with an initial Sandwich approximation and refines it by successively taking additional knots on the boundary of P to define the approximation. Different versions of the Sandwich algorithm differ in the way how the next point is chosen.

In a Sandwich approximation, the boundary of the convex figure P is naturally divided into pieces by the knots. Since the effect of inserting a new knot is to divide one of these pieces into two, we refer to the strategy for selecting the next knot as the *partition rule*. We consider four different partition rules, that would naturally come to one's mind (interval bisection, slope bisection, maximum error rule, and chord rule), and show that they all lead to Sandwich algorithms where the global approximation error decreases by the order of $O(1/n^2)$.

Motivation of the problem. A very closely related and equally important problem is that of approximating a convex function of one variable by a piecewise linear function. This problem is essentially equivalent to the approximation of a convex figure.

The applications for approximation of convex bodies (or convex functions) can be classified into two categories:

1. It is computationally expensive to determine a point on the boundary of P, and we want to get an approximate idea of the overall shape of P. This occurs for example if one wants to determine the efficient point curve of bicriteria linear programs or the solution of parametric problems.

2. The body P is completely known, but it is nevertheless expedient to replace it by a polygon with few vertices, as such a polygon might be easier to handle. In fact, the complexity of any problem in computational geometry that involves polygons depends on the size of the polygons involved, and in many cases (for example in motion planning) the dependence on

Author's address: University of Waterloo, Department of Combinatorics and Optimization, Waterloo, Ontario, Canada N2L 3G1. On leave from Institut für Mathematik, Technische Universität Graz, Kopernikusgasse 24, A-8010 Graz, Austria (until August 1990). Partial support by the Fonds zur Förderung der wissenschaftlichen Forschung, projects S32/01 and P7486-PHY, is gratefully acknowledged.

the number of vertices is quadratic or of even higher degree. In such cases, replacing the input of a problem by a simpler approximation is a way to get approximate results while speeding up the calculations. (A different approximation problem from ours, which was motivated in this way, has been considered in Fleischer et al. [1990] [1991].)

In the first case, the problem is to get an acceptable approximation with as few points ("function evaluations") as possible. Here the Sandwich algorithm is the ideal candidate, because the Sandwich approximation is just the best approximation that can be obtained by using the information that is known. In the second case, the approximation problem can often be solved by more direct methods (cf. Imai and Iri [1986], McClure and Vitale [1975], or Cantoni [1971]). However, these methods are sometimes complicated, and it is often not even clear what the optimization criterion of the approximation should really be. Thus, the Sandwich algorithm might still be the method of choice if a simple and fast algorithm with a good performance guarantee is asked for.

The error measure and the partition rules. The error measure that we consider in this paper is the maximum distance between the inner and the outer approximation (the Hausdorff distance)

$$\max_{x \in P_{\text{outer}}} \min_{y \in P_{\text{inner}}} \operatorname{dist}(x, y),$$

where $\operatorname{dist}(x, y)$ denotes the Euclidean distance. We shall actually first derive our results for convex *functions.* In this case, we consider the maximum *vertical* distance between the lower and the upper approximation. Other error measures can also be handled.

Now we describe the Sandwich algorithm in more detail, for the case of approximating a convex function (see figure 2): An initial approximation is obtained by evaluating the function and its derivatives at the endpoints of the definition interval. At any stage during the algorithm, we have selected some knots and thus subdivided the original interval into a number of subintervals. We compute the maximum error in each subinterval and select the interval with the largest error. This interval is the further subdivided by inserting an additional knot as specified by the partition rule. This process is continued for a given number of iterations or until a specified error bound is met.

The partition rules that we consider are as follows (cf. figure 3):

(i) *The interval bisection rule:* the interval is partitioned into two equal parts.

(ii) *The slope bisection rule:* We find the supporting line whose slope is the mean value of the slopes of the tangents at the endpoints. We partition the interval at the point (at some point) where this line touches the function.

(iii) *The maximum error rule:* The interval is partitioned at the breakpoint of the lower approximation, i. e., at the point where the error between the two approximations is largest.

(iv) *The chord rule* is similar to the slope bisection rule. However, we take the slope of the line connecting the two endpoints as the slope of the supporting line.

The rules fall into two classes: Rules (i) and (iii) specify the *abscissa* of the new point, whereas rules (ii) and (iv) find the point by specifying the *slope* of a supporting line. Which way of specifying the new breakpoint is more convenient depends on the application. The chord rule is actually also a kind of maximum error rule, since it selects the point *on the function* whose distance from the upper approximation is maximum.

Previous results. The problem of approximating a convex body by a polygon (or polytope) has attracted a great deal of attention in the theoretical literature (cf. the survey of Gruber [1983]). It is well known that the distance between a convex body and its best approximating n-gon is $O(1/n^2)$. It is easy to see (by considering the case of a circle) that this convergence rate is best possible. Most of these results are existential in nature. A proof of the quadratic convergence rate of the Sandwich algorithm for the interval bisection rule was given in Sonnevend [1984] (in a more general setting), and for the case of the interval bisection rule and the slope bisection rule in Burkard, Hamacher, and Rote [1990] (with a different proof). Our results are a quantitative

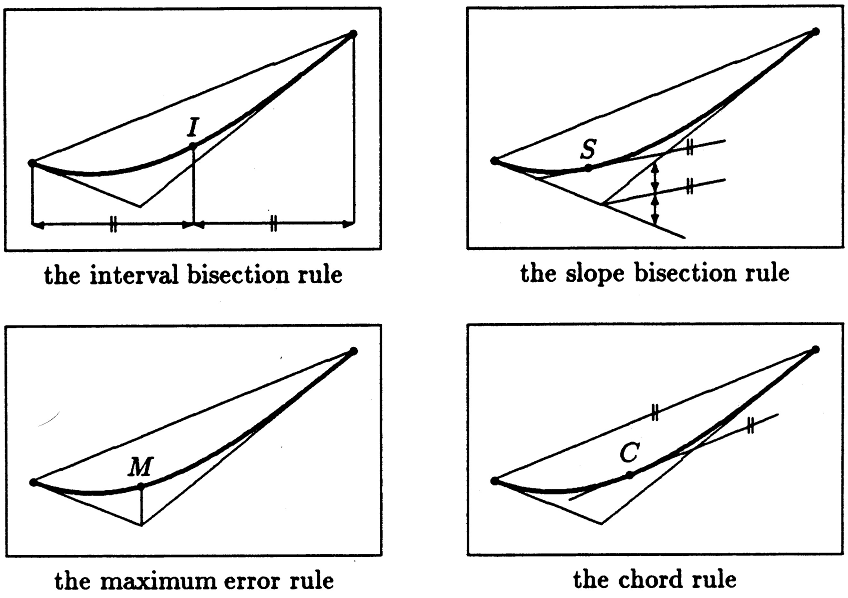

Figure 3: Four partition rules for the Sandwich algorithm

counterpart to the results on *probing* of polygons (cf. Cole and Yap [1987]), where a polygon is to be reconstructed by asking, for example, for the intersection of the polygon with a specified line or for the supporting line with a specified direction.

Geometric duality of convex functions. We can describe a convex function $h: [a, b] \to \mathbb{R}$ as the set of pairs $((p, q), (k, d))$, where (p, q) is a point on the graph of the function, i. e., $q = h(p)$, and $y = kx + d$ is a supporting line in this point, i. e., it contains the point (p, q) and no point of the function lies below it. The dual transformation $\mathcal{D}$ maps the pairs $((p, q), (k, d))$ to the pairs $((k, -d), (p, -q))$. This transformation is a special projective duality (cf. Edelsbrunner [1987], section 1.4 or 15.2). It transforms the given function h into another convex function $\mathcal{D}(h)$.

Applying the interval bisection rule to a function corresponds to applying the slope bisection rule to its dual, and it is easy to check that the vertical errors are the same in both cases. Similarly, the maximum error rule and the chord rule are dual to each other.

Results — the interval and slope bisection rules

Theorem 1 (Theorem 2.3 of Burkard, Hamacher, and Rote [1990].) *Suppose we are given a function h defined on an interval $[a, b]$ of length $L = b - a$, where the function values and the one-sided derivatives $h^+(a)$ and $h^-(b)$ have been evaluated at the endpoints a and b. Let the slope difference be $\Delta = h^-(b) - h^+(a)$. Then, in order to make the greatest vertical error between the upper and the lower approximation smaller than or equal to ε, the interval bisection rule or the slope bisection rule needs at most $z_\varepsilon(L\Delta)$ additional evaluations of $h(x)$, $h^-(x)$, and $h^+(x)$, where*

$$z_\varepsilon(L\Delta) = \begin{cases} 0, & \text{for } L\Delta/\varepsilon \le 4, \\ \left\lceil \sqrt{\dfrac{9}{8}\dfrac{L\Delta}{\varepsilon}} - 2 \right\rceil, & \text{for } L\Delta/\varepsilon > 4. \end{cases}$$

Corollary 1 *If we always subdivide the interval with largest error according to the interval bisection rule or according to the slope bisection rule, then the maximum vertical error after $M \ge 2$ evaluations of h, h^-, and h^+ is at most*

$$\frac{9}{8}\frac{L\Delta}{M^2}.$$

Results — the maximum error rule and the chord rule

Theorem 2 *Suppose we are given a function h defined on an interval $[a,b]$ of length $L = b - a$, where the function values and the one-sided derivatives $h^+(a)$ and $h^-(b)$ have been evaluated at the endpoints a and b. Let the slope difference be $\Delta = h^-(b) - h^+(a)$. Then, in order to make the greatest vertical error between the upper and the lower approximation smaller than or equal to ε, the maximum error rule or the chord rule needs at most $m_\varepsilon(L\Delta)$ additional evaluations of $h(x)$, $h^-(x)$, and $h^+(x)$, where*

$$m_\varepsilon(L\Delta) = \begin{cases} 0, & \text{for } L\Delta/\varepsilon \le 4, \\ \left\lceil \sqrt{\frac{L\Delta}{\varepsilon}} - 2 \right\rceil, & \text{for } L\Delta/\varepsilon > 4. \end{cases}$$

Corollary 2 *If we always subdivide the interval with largest error according to the maximum error rule or to the chord rule, then the greatest vertical error after $n \ge 2$ evaluations of h, h^-, and h^+ is at most*

$$L\Delta/n^2.$$

Approximation of convex plane figures

Theorem 3 *With $n \ge 4$ knots, the Sandwich algorithm approximates a convex plane figure P of circumference D with an error at most $9D/(n-2)^2$ in case of the interval or slope bisection rule, or at most $8D/(n-2)^2$ in case of the maximum error rule or the chord rule.*

References

Rainer E. Burkard, Horst Hamacher, and Günter Rote [1990]
Approximation of convex functions and applications to separable convex programming, technical report, Technische Universität Graz, Institut für Mathematik (submitted for publication).

A. Cantoni [1971]
Optimal curve fitting with piecewise linear functions, *IEEE Transactions on Computers* **C-20**, 59-67.

Richard Cole and Chee-Keng Yap [1987]
Shape from probing, *J. Algorithms* **8**, 19–38.

Herbert Edelsbrunner [1987]
Algorithms in Combinatorial Geometry, Springer-Verlag, Berlin etc.

Rudolf Fleischer, Kurt Mehlhorn, Günter Rote, Emo Welzl, and Chee-Keng Yap [1990]
On simultaneous inner and outer approximation of shapes, in: *Computational Geometry*, Proc. Sixth Ann. Symp. Comput. Geom., Berkeley, California, June 6-8, 1990, Association for Computing Machinery, pp. 216–224.

Rudolf Fleischer, Kurt Mehlhorn, Günter Rote, Emo Welzl, and Chee-Keng Yap [1991]
Simultaneous inner and outer approximation of shapes, submitted to *Algorithmica*.

Bernd Fruhwirth, Rainer E. Burkard, Günter Rote [1989]
Approximation of convex curves with application to the bicriterial minimum cost flow problem, *European Journal of Operational Research* **42**, 326–338.

Peter M. Gruber [1983]
Approximation of convex bodies, in: Peter M. Gruber and Jörg M. Wills (eds.): *Convexity and its applications*, Birkhäuser, Basel, Boston 1983, pp. 131–162.

Hiroshi Imai and Masao Iri [1986]
An optimal algorithm for approximating a piecewise linear function, J. Information Processing **9**, 159–162.

George Martelli [1962]
Jemmy Twitcher — A Life of the Fourth Earl of Sandwich, 1718–1792, Jonathan Cape, London.

D. E. McClure and R. A. Vitale [1975]
Polygonal approximation of plane convex bodies, *J. Math. Anal. Appl.* **51**, 326–358.

György Sonnevend [1984]
Sequential, optimal order algorithms for the uniform approximation of functions with convex r-th derivatives, *Analysis Mathematica* **10**, 311-335.