Equidistribution in BEM based on local and global error estimates

by

R.M.M. Mattheij and K. Wang
Equidistribution in BEM based on local and global error estimates
R.M.M. Mattheij and K. Wang
Department of Mathematics and Computing Science,
Eindhoven University of Technology, The Netherlands

Abstract
In this paper we analyse adaptivity for BEM. It is based on estimating the local or global errors and employing equidistribution. Based on this we describe two possibilities to obtain an approximate solution, both on a equidistributed grid and such that the error is below a given required tolerance.

1 Introduction
An important problem in solving differential equations numerically is the complexity of the computations. Directly related to this is the efficiency of the way one finds a grid, sufficiently accurate to keep errors below certain tolerance. For this latter problem, not only undershoot (i.e. too few points) is undesired, but also overshoot (i.e. too many points). There exists a host of strategies to deal with this question in general. In BEM this problem has attracted more attention recently (cf. [1]). Of course, the crucial point is how one can find an “optimal” grid by using information gathered during the computation. This leads to adaptive gridding methods.

In this paper we consider adaptive gridding both of the local error and of the global error. The general idea is to use equidistribution that employs knowledge of the behaviour of these errors, as well as a good error estimator. As mentioned such technique needs to be fairly inexpensive, if only because it is likely to be used in an iterative setting. Since we mainly intend to explain the method principle, we will restrict ourselves to Laplace problems in 2-D. In principle the argumentation can be generalised. For example, one can use the same method in solving Stokes equation and heat problems (cf. [10]).

The paper is built up as follows. In Section 2 we give the problem setting, formulating the problem as a matrix-vector equation. Next we outline what we mean by equidistribution in Section 3. The two cases of equidistribution, viz. for local errors and global errors, are considered in Section 4 and Section 5 respectively. In particular we show how one can estimate these errors from computed data. A variety of approaches can be used, based on this error estimating and equidistributing. In Section 6 we give two possibilities. Finally, in Section 7 we illustrate our analysis by some examples.

2 General setting
Let us consider a simple Laplace equation on a domain \( \Omega \subset \mathbb{R}^2 \)

\[
\nabla^2 u(x) = 0, \quad x \in \Omega.
\]

(1)
Denote by $\Gamma$ the boundary $\partial \Omega$, to be split into two parts $\Gamma_u$ and $\Gamma_q$, with $\Gamma = \Gamma_u \cup \Gamma_q$, $\Gamma_u \cap \Gamma_q = \emptyset$. Let the following boundary condition be given on $\Gamma$

\[
\begin{align*}
n & = \tilde{u}, & \text{on } \Gamma_u, \\
q & = \partial u / \partial n = \tilde{q}, & \text{on } \Gamma_q.
\end{align*}
\]

From this we derive the boundary integral equation

\[c(x)u(x) + \int_{\Gamma} q^*(x,y)u(y)d\Gamma y - \int_{\Gamma} u^*(x,y)q(y)d\Gamma y = 0, \quad x \in \Gamma,
\]

where $u^*$ is the fundamental solution, $q^*$ is the normal derivative of $u^*$ with respect to $y$, and $c$ depends on the smoothness of $\Gamma$ (usually it is $1/2$).

Introducing the single layer potential $K^s$ and double layer potential $K^d$ by

\[
K^s u(x) := \int_{\Gamma} u^*(x,y)u(y)d\Gamma y,
\]
\[
K^d u(x) := \int_{\Gamma} q^*(x,y)u(y)d\Gamma y,
\]

we can write (3) as

\[(cI + K^d)u - K^s q = 0.
\]

The boundary condition (2) induces a characteristic function $\chi$ on $\Gamma$ such that

\[
\chi u = \begin{cases} 
  u, & x \in \Gamma_u, \\
  0, & x \in \Gamma_q.
\end{cases}
\]

Using (6) we see that we can rewrite (5) as

\[c(1 - \chi)u + K^d(1 - \chi)u - K^s q = -c\chi u - K^d\chi u + K^s(1 - \chi)q.
\]

Introducing the unknown

\[w := (1 - \chi)u + \chi q,
\]

and using (2) in (4) we see that we can simply rewrite (7) as

\[Lw(x) = f(x).
\]

Let us now split up $\Gamma$ into $N$ elements, $\Gamma_1, \cdots, \Gamma_N$ with grid points $x_1, \cdots, x_N$ ($\Gamma_i$ has end points $x_i$ and $x_{i+1}$, and $x_{N+1} := x_1$). We now discretize (9) using piecewise linear polynomials $\phi_j$, $j = 1, \cdots, N$, say. The matrices resulting from the single layer and double layer potential, $H$ and $G$ respectively are then given by

\[
H = (k_{ij}^s), \quad G = (k_{ij}^d).
\]
For the elements we have
\[ k_{ij}^e := \int_{\Gamma} u^*(x_i, x) \phi_j(x) d\Gamma = \sum_{e=1}^{N} k_{ij}^e, \] (11)
where \( k_{ij}^e := \int_{\Gamma_e} u^*(x_i, x) \phi(x) d\Gamma, \)
and
\[ k_{ij}^d := \int_{\Gamma} q^*(x_i, x) \phi_j(x) d\Gamma = \sum_{e=1}^{N} k_{ij}^d, \] (12)
where \( k_{ij}^d := \int_{\Gamma_e} q^*(x_i, x) \phi(x) d\Gamma. \)

So we obtain
\[ H(I - Q)u - Gq = -QHu + (I - Q)Gq, \] (13)
where \( Q \) is the projection of the grid function on \( \Gamma_u. \) The latter equation gives approximate values of \( u \) at \( \Gamma_q \) and of \( q \) at \( \Gamma_u \) that can be used, together with the basis polynomials, to produce an approximation of \( w, \) \( w_h \) say. Substituting \( w_h \) in (9) yields a residual \( r_h, \) i.e.
\[ \mathcal{L}w_h(x) = f(x) + r_h(x). \] (14)
This residual is also called the local error.

The global error, defined as
\[ e_h(x) := w(x) - w_h(x), \] (15)
can be seen to satisfy (see (9) and (14))
\[ \mathcal{L}e_h(x) = r_h(x). \] (16)

We next discuss how we compute or estimate \( r_h \) and \( e_h. \)

### 3 Estimates for the local error

Before proceeding to discuss how we will estimate \( r_h, \) we first remark that we do not need to know \( r_h \) very accurately, if only the approximation reflects the proper order of accuracy.

Let \( u^j \) denote a nodal value so that the numerical solution \( u_h(x) \) can be written as
\[ u_h(x) = \sum_{j=1}^{N} u^j \phi_j(x). \] (17)

Then we apparently have from (13) for a typical point, \( \bar{x} \in \Gamma_i \) say,
\[ u_h(\bar{x}) = \frac{1}{c(\bar{x})} \left[ \sum_{j=1}^{N} k^*_j(\bar{x}) - \sum_{j=1}^{N} k^d_j(\bar{x}) \right], \] (18)
where
\[
kJ(x) := \int_{\Gamma} \ln \|\hat{x} - x\| \phi_j(x) d\Gamma = \sum_{i=1}^{N} \int_{\Gamma_i} \ln \|\hat{x} - x\| \phi_j(x) d\Gamma,
\]
\[
kj(x) := \int_{\Gamma} \frac{\ln \|\hat{x} - x\|}{\|\hat{x} - x\|^p} \phi_j(x) d\Gamma = \sum_{i=1}^{N} \int_{\Gamma_i} \frac{\ln \|\hat{x} - x\|}{\|\hat{x} - x\|^p} \phi_j(x) d\Gamma.
\]

Our first choice is that we take \( \hat{x} \) as the point in the middle of \( \Gamma_i \); the rationale for this is that we expect the errors there to be the largest on average.

The second choice is that we use the elements of \( H \) and \( G \) to find \( k_j(\hat{x}) \) and \( k_j(x) \) within approximation error accuracy. Let us demonstrate this for \( k_j(\hat{x}) \). Let \( \Gamma_i \) be sufficiently far from \( \Gamma_i \), then we have

\[
\ln \|\hat{x} - x\| - \ln \|x_i - x\| = \frac{\|\hat{x} - x\| - \|x_i - x\|}{\|x_i - x\|} + O\left(\|\hat{x} - x\| - \|x_i - x\|\right)^2). \tag{20}
\]

Since \( \|\hat{x} - x\| - \|x_i - x\|\) < \( h_i \), \( \|x_i - x\| > dist(\Gamma_i, \Gamma) \) (minimal distance between points of \( \Gamma_i \) and \( \Gamma \)) and \( \|\phi_j\| \) bounded on \( \Gamma_i \), say \( \|\phi_j\| < C \), we find from (20)

\[
\int_{\Gamma_i} \ln \|\hat{x} - x\| \phi_j(x) d\Gamma - \int_{\Gamma_i} \ln \|x_i - x\| \phi_j(x) d\Gamma \leq \frac{Ch_i h_i}{dist(\Gamma_i, \Gamma)} + O(h_i^2). \tag{21}
\]

So we conclude that

\[
\int_{\Gamma_i} \ln \|\hat{x} - x\| \phi_j(x) d\Gamma = \int_{\Gamma_i} \ln \|x_i - x\| \phi_j(x) d\Gamma + O(h_i^2). \tag{22}
\]

If the elements are close, say if \( l = i - 1, i \) or \( i + 1 \), we have to compute \( \int_{\Gamma_i} \ln \|\hat{x} - x\| \phi_j(x) d\Gamma \). Bearing in mind our earlier remark about values that only have to be approximately correct, we thus only do this for \( l = i - 1, i \), and \( i + 1 \). We find

\[
kj(\hat{x}) = \sum_{l=1}^{N} k_{ij} + \sum_{l=i-1}^{i+1} \int_{\Gamma_i} \ln \|\hat{x} - x\| \phi_j(x) d\Gamma. \tag{23}
\]

In view of (20) this is a second order approximation. Note that the computation of \( r_h(\hat{x}) \) for \( \hat{x} \) being the middle point of \( \Gamma_i \), \( i = 1, \cdots, N \) is an \( O(N) \) computation. Direct evaluation would require the evaluation of \( O(N^2) \) integrals.

4 Estimates for the global error

Alternatively we may estimate the global error directly. Let us define the interpolation operator \( \mathcal{P}_h \) which maps a function on the boundary to a piecewise interpolation polynomial with basis \( \{\phi_j\} \). Then we have for \( w \) as defined in Section 2

\[
w - w_h = (w - \mathcal{P}_h w) + (\mathcal{P}_h w - w_h) = (w - \mathcal{P}_h w) + \mathcal{P}_h (w - w_h). \tag{24}
\]
It can be shown that (cf. [3])
\[
\frac{\|P_h(w - w_h)\|}{\|w - w_h\|} \rightarrow 0, \quad \text{as } h \rightarrow 0.
\]
(25)

As a consequence we find, asymptotically
\[
e_h := w - w_h \approx w - P_h w.
\]
(26)

This means that the interpolation error is the dominant part of the discretisation error. Of course we do not know \(w\). However once we have found a good approximation of \(w\) we may still use (26) by applying it to estimating the interpolation error based on \(w_h\) instead.

5 Equidistribution

With these estimates of the error we can now employ equidistribution. For this it is important to know accuracy order. The idea is the following (cf. [2]). Let the error on \(\Gamma_i\) be given by \(F_i h_i^p\) (\(p\) is the order, \(h_i\) the length of \(\Gamma_i\) and \(F_i\) depends on the smoothness of the solution). If the error at all elements is the same, i.e. \(F_i h_i^p\) is independent of \(i\) then we call it equally distributed, i.e. for some constant \(E\)
\[
F_i h_i^p = E.
\]
(27)

We can work this out to see that this implies for \(h_i\)
\[
F_i^{1/p} h_i = E^{1/p} =: K,
\]
(28)

or
\[
h_i = K F_i^{-1/p}.
\]
(29)

Note that \(E\) should be lower than the required error tolerance to have sufficiently small \(h_i\).

More elegantly this can be formulated using a so called monitor function
\[
\psi(x) := F(x)^{1/p},
\]
(30)

which represents the "(non)smoothness" of the solution as a function of \(x\). The continuous analogue of (28) then reads,
\[
\int_{\Gamma_i} \psi(x) d\Gamma =: K,
\]
(31)

where, ideally, \(K = (TOL)^{1/p}\), if \(TOL\) denotes the required tolerance.

We now use this in combination with the local error as follows. If we approximate (31) by the midpoint rule we obtain
\[
K \approx \psi(\bar{x}_i) h_i,
\]
(32)
where \( \hat{x}_i \) is the midpoint of \( \Gamma_i \). Now we can compute from (18) the residual \( r_h(\hat{x}_i) \). Clearly as an estimate of \( F_i \) we then take \( |r_h(\hat{x}_i)|/h_i^2 \) (which should be independent of \( h_i \)). Consequently we use the monitor function \( \psi \) defined by

\[
\psi(\hat{x}_i) = \frac{1}{h_i} |r_h(\hat{x}_i)|^{1/2},
\]

whence

\[
K = |r_h(\hat{x}_i)|^{1/2}.
\]

If we have not reached equidistribution we will rather find

\[
K_i := |r_h(\hat{x}_i)|^{1/2},
\]

not be constant. Identifying local and global errors (approximately true for a well conditioned problem) we thus have

\[
\max_i |e_i| \approx \max_i |r_h(\hat{x}_i)| = \max_i K_i^2. \tag{36}
\]

We can also use this equidistribution for the global error. To do this let us first produce a cubic spline \( \bar{w} \) say with \( X_j \) as knots and \( w_h(x_j) \) as function values. Anticipating a second order error in \( w_h(x_j) \), we simply use \( \bar{w} \) to approximate \( w - P_h w \). On \( \Gamma_i \) we find

\[
\frac{|w - P_h w|}{|\bar{w} - P_h \bar{w}|} \approx \frac{1}{2} \int \frac{d^2 \bar{w}(s)}{d \delta^2} \left| \frac{d^2 \bar{w}(s)}{d \delta^2} \right| ds\]
\[
= \frac{1}{2} \int \frac{d^2 \bar{w}(s)}{d \delta^2} \left| \frac{d^2 \bar{w}(s)}{d \delta^2} \right| ds\]
\[
\approx \frac{1}{\delta h_i^2} \left| \frac{d^2 \bar{w}(s)}{d \delta^2} \right| \tag{37}
\]

Here \( \delta \) is the arclength parameter. Since \( \frac{d^2 \bar{w}(s)}{d \delta^2} \) can be found explicitly (and within \( O(h_i^2) \) accuracy), we see that the second order character of the error induces a monitor function \( \psi \) given by

\[
\psi(x) := \left( \left| (1 - \chi) \frac{d^2 \bar{w}(s)}{d \delta^2} \right| + \left| \chi \frac{d^2 \bar{w}(s)}{d \delta^2} \right| \right)^{1/2}, \quad x = x(s) \tag{38}
\]

Again we conclude that

\[
\|e_h\| \approx \max_i K_i^2. \tag{39}
\]

6 Algorithms and numerical examples

From the results of Section 3 and Section 4 we can derive estimates for the local or global errors. There is a variety of ways how this can be employed to achieve both equidistribution and sufficient accuracy. In general nothing is known about the solution. So suppose we are given \( N \) equi-spaced points (measured along the boundary curve) \( x_1^0, \ldots, x_N^0 \).
Let \( u^0 \) and \( q^0 \) be the computed discrete solution (\( h^0 \) is used as a generic indication for the initial mesh, \( h^0, \ldots, h^n \) say).

At the \( i \)th element \( \Gamma_i \), say, we compute \( \phi(x) \). Let \( K^Q_i \) be defined by (cf. (35))

\[
K^Q_i := \int_{\Gamma_i} \phi(x) d\Gamma.
\]

Clearly

\[
\sum_{i=1}^N K^Q_i = \int_{\Gamma} \phi(x) d\Gamma.
\]

The best guess for an optimal “share” per interval is then

\[
L^0 := \sum_{i=1}^N K^Q_i / N.
\]

This induces a new set of points \( x^1_i, \ldots, x^N_i \) say (with elements \( \Gamma^1_i \) having \( x^1_i \) and \( x^i_{i+1} \) as end points with \( x^N_i = x^0_i \))

\[
\int_{x^0_i}^{x^i_{i+1}} \psi(x) d\Gamma = \int_{\Gamma^1_i} \psi(x) d\Gamma = L^0.
\]

Hence we choose \( x^1_i \) sequentially with \( x^1_1 = x^0_1 \). Once the new grid has been found, we can perform the BEM simulation again. Obviously we can now compute a set \( \{K^1_i\} \) and thus an \( L^1 \) and repeat the process until the \( \{K_i\} \) are approximately equal to one another.

**Example 1:** Consider a Dirichlet problem on a circle with radius \( r = 2 \). The exact solution is \( u = \log \|x - x_0\| \) with \( x_0 = (-2, 0)^T \). As can be seen the normal derivative has a large gradient along the boundary in the neighbourhood of \( (-2, 0)^T \). We performed the computations with 30 elements having equal length initially. Using local error control we obtained a nicely distributed grid. In Figure 1 we give \( q \) (the normal derivative) as a function of \( \theta \). The solid line represents the exact solution.

**Example 2:** Consider the problem with boundary data as shown in Figure 2. The normal derivative has a large gradient on the topside. The initial mesh has 10 equi-sized elements on \( AB, BC \) and \( DA \) each, 30 equi-sized elements on \( CA \). The normal derivatives on the topside are shown shown in Figure 3. The solid line here and in subsequence figures represents the “exact” solution obtained by using 2048 equi-spaced elements. In this example we used global error control.

Of course, in order to retain the actually required accuracy, we may have to refine. Suppose we have an error tolerance \( TOL \) and an estimate \( E \). If \( E \) is smaller than \( TOL \) we may have an overshoot, but accept the mesh anyway. If \( E \) is much larger than \( TOL \) we have to refine the mesh. Employing the \( p \)th order character we may choose a new mesh-width, \( h^\text{new} \) say, instead of the old, \( h^\text{old} \) say, by

\[
h^\text{new} = h^\text{old} \left( \frac{TOL}{E} \right)^{1/p}.
\]
The actual number of new elements in \( \Gamma_i \), \( N_i \) say, is then taken equal to

\[
N_i = \lceil \left( \frac{E}{TOL} \right)^{\frac{1}{p}} \rceil.
\]  

(45)

**Example 3:** In this example we solve the same problem as in Example 2, but we require that the accuracy should satisfy a prescribed \( TOL = 0.01 \). The initial grid is the same as that in Example 2: 10 equi-sized elements on \( AB \), \( BC \) and \( DA \) each, 30 equi-sized elements on \( CA \). After 2 iterations the process stops. The results are shown in Figure 4. Each iteration consists of equidistribution and tolerance control. The errors contributed by every element are all between 0.0085 and 0.0107, the average is 0.0098. This is what we expect.
7 Conclusions

In this paper we have considered methods to find optimal grids based on equidistributing the local error or global error. Furthermore we have shown how to find the optimal grid in terms of a given tolerance. Numerical examples illustrate these.
Figure 4: (Example 3) Equidistributing with tolerance control

References


