The Hybrid Galerkin Boundary Element Method

Ivan G. Graham *, Wolfgang Hackbusch ** and Stefan A. Sauter **

* *School of Mathematical Sciences, University of Bath, Bath BA2 7AY, United Kingdom.* igg@maths.bath.ac.uk.

** *Praktische Mathematik, Christian-Albrechts-Universität zu Kiel 24098 Kiel, Germany.*
wh@numerik.uni-kiel.de, sas@numerik.uni-kiel.de

Abstract

In the implementation of boundary element methods for 3D elliptic PDEs in a given geometry there are two main numerical tasks: the stiffness matrix assembly and the solution of the resulting linear systems. The linear solve (performed directly) is $O(n^3)$ - where n is the number of degrees of freedom - and is asymptotically the most expensive component. Hence there are a number of modern techniques (multipole, panel-clustering, wavelets) for reducing this cost. However for *practical* discretisations it is the formation of the stiffness matrix which often dominates (even though asymptotically it is only $O(n^2)$). Saving in the matrix assembly process may sometimes be achieved by replacing the variational Galerkin method with the collocation method. However, this is not a panacea. In some cases it is not clear what an appropriate set of collocation points should be and in a great many more cases the collocation method has no theoretical justification. The last point is not just a mathematical nicety - a relevant convergence theory often leads to better practical techniques, such as distinguishing the best norms on which to base an adaptive procedure. In this paper we describe the first results on a new quadrature method for computing the Galerkin stiffness matrix arising from piecewise linear finite elements on surfaces in $\mathbb{R}^3$. This method in effect approximates the true Galerkin matrix away from the diagonal by appropriately averaged elements of a related discrete collocation (or *Nyström*) matrix and hence uses relatively few kernel evaluations. This "hybrid" Nyström - Galerkin method can be shown to be stable and to enjoy the same convergence rate in the energy norm as the Galerkin method itself. Numerical tests show that the method produces solutions which are qualitatively as good as the Galerkin method, but at substantially reduced cost. This paper is the first to describe this new method and will do so only in the context of a single model example. Generalisations and proofs of the theoretical results described here will be contained in the later paper Graham, Hackbusch & Sauter (1997).

The Galerkin Boundary Element Method

For this discussion we shall restrict attention to the piecewise linear approximation of the single layer potential equation arising from the Laplace operator:

$$\mathcal{K}u(\boldsymbol{x}) := \int_\Gamma k(\boldsymbol{x},\boldsymbol{y})u(\boldsymbol{x})dx = f(\boldsymbol{x}), \tag{1}$$

where Γ is a smooth closed bounded surface in $\mathbb{R}^3$ and $k(\boldsymbol{x},\boldsymbol{y}) = 1/|\boldsymbol{x}-\boldsymbol{y}|$.

We shall assume that Γ is specified by the provision of an interpolating polyhedron $\tilde{\Gamma}$ and a continuous map $\boldsymbol{\eta} : \tilde{\Gamma} \to \Gamma$. The map $\boldsymbol{\eta}$ is assumed smooth on each of the faces $\tilde{C}_i$ of $\tilde{\Gamma}$ and its restriction $\boldsymbol{\eta}_i : \tilde{C}_i \to \boldsymbol{\eta}(\tilde{C}_i)$ is assumed invertible for each i. The surface Γ is then assumed to be partitioned by the images $\cup_i\{\boldsymbol{\eta}(\tilde{C}_i)\}$ and such a situation is typical in geometrical modelling of surfaces: The $\boldsymbol{\eta}_i$ could be constructed from locally defined suitably high order interpolants of Γ.

We discretise (1) with piecewise linear basis functions in a standard way. First we triangulate $\tilde{\Gamma}$ with a conforming triangular mesh. This induces a curvilinear triangular mesh on Γ via $\boldsymbol{\eta}$. Let $\mathcal{N}$ denote the nodes on Γ and let h denote the mesh diameter. Each node $\boldsymbol{x}_p \in \mathcal{N}$ has a preimage $\tilde{\boldsymbol{x}}_p \in \tilde{\Gamma}$ and we define the standard basis functions ϕ_p on Γ by requiring that

$$\phi_p(\boldsymbol{x}) = \tilde{\phi}_p(\tilde{\boldsymbol{x}}) , \quad \boldsymbol{x} \in \Gamma ,$$

where $\tilde{\phi}_p$ is the usual piecewise linear hat function defined on the conforming mesh on $\tilde{\Gamma}$ and where $\boldsymbol{\eta}(\tilde{\boldsymbol{x}}) = \boldsymbol{x}$.

If we then set $\mathcal{S}_h = \text{span}\{\phi_p : \boldsymbol{x}_p \in \mathcal{N}\}$, and let $(\cdot,\cdot)$ denote the usual L_2 inner product on Γ, the **Galerkin** method for (1) is to seek $u_h \in \mathcal{S}_h$ such that

$$(\mathcal{K}u_h, \phi_p) = (f, \phi_p) , \quad \boldsymbol{x}_p \in \mathcal{N} . \tag{2}$$

It is well-known that u_h is well-defined and that, if u is sufficiently smooth, then $\|u-u_h\|_{-1/2} \le Ch^{5/2}$ as $h \to 0$, for some constant C, where $\|\cdot\|_{-1/2}$ is the norm in the Sobolev space $H^{-1/2}$, which is equivalent to the energy norm for this problem, namely $\|v\|_a := (\mathcal{K}v, v)^{1/2}$.

This paper concerns the set-up time for the stiffness matrix $K = (K_{pq})$ arising from (2). By expressing u_h as a linear combination of the ϕ_q this is given by

$$K_{pq} = \int_\Gamma\int_\Gamma k(\boldsymbol{x},\boldsymbol{y})\phi_q(\boldsymbol{y})\phi_p(\boldsymbol{x})dydx , \quad \boldsymbol{x}_p,\ \boldsymbol{x}_q \in \mathcal{N} . \tag{3}$$

For each p, the support of ϕ_p, denoted $\text{supp}(\phi_p)$, consists of the curvilinear triangles meeting at $\boldsymbol{x}_p$ and so

$$K_{pq} = \sum_{T\in\text{supp}(\phi_p)} \sum_{T'\in\text{supp}(\phi_q)} \int_{T\times T'} k(\boldsymbol{x},\boldsymbol{y})\phi_q(\boldsymbol{y})\phi_p(\boldsymbol{x})dydx . \tag{4}$$

In practice the integrals K_{pq} are replaced by quadrature approximations $\tilde{K}_{pq}$, yielding a discrete Galerkin solution $\tilde{u}_h$. In Graham, Hackbusch & Sauter (1997) we prove the following result.

Lemma 1 Suppose that the meshes on Γ are refined in a quasi-uniform way. Suppose also that

$$|K_{pq} - \tilde{K}_{pq}| \le Ch^3 \mathcal{A}(\mathrm{supp}(\phi_p))\mathcal{A}(\mathrm{supp}(\phi_q)) \ , \quad \text{as } h \to 0 \tag{5}$$

(with $\mathcal{A}$ denoting area). Then the discrete Galerkin solution $\tilde{u}_h$ is well-defined and satisfies the estimate $\|u - \tilde{u}_h\|_{-1/2} \le Ch^{5/2}$ as $h \to 0$. $\square$.

To satisfy the key requirement (5), note that the individual entries in the summation (4) may be written

$$\begin{aligned} &\int_{T\times T'} k(\boldsymbol{x},\boldsymbol{y})\phi_q(\boldsymbol{y})\phi_p(\boldsymbol{x})dydx \\ &= \int_{\tilde{T}\times\tilde{T}'} k(\boldsymbol{\eta}(\tilde{\boldsymbol{x}}),\boldsymbol{\eta}(\tilde{\boldsymbol{y}}))g(\tilde{\boldsymbol{x}})g(\tilde{\boldsymbol{y}})\tilde{\phi}_q(\tilde{\boldsymbol{y}})\tilde{\phi}_p(\tilde{\boldsymbol{x}})d\tilde{y}d\tilde{x} \quad , \end{aligned} \tag{6}$$

where the integral is now over the product of a pair of plane triangles $\tilde{T}$, $\tilde{T}'$ in $\tilde{\Gamma}$ and $dx = g(\tilde{\boldsymbol{x}})d\tilde{x}$ when $\boldsymbol{x} = \boldsymbol{\eta}(\tilde{\boldsymbol{x}})$. The integrals (6) can be divided into two classes: *singular integrals*, in which $\tilde{T}, \tilde{T}'$ touch or coincide and the remainder which we call *nonsingular integrals*.

The singular integrals may be reduced to sums of smooth integrals by transformation techniques Hackbusch & Sauter (1993), Sauter (1996), Erichsen & Sauter (1996), and these can then be approximated to arbitrary accuracy by Gauss rules. Formulae for the degrees of precision required to ensure (5) can be deduced from Sauter (1996).

When $\mathrm{supp}(\phi_p)$ does not touch $\mathrm{supp}(\phi_q)$, which is the more common case as $h \to 0$, all the integrals in (4) are nonsingular. Each individual integral (6) may then be approximated using the cross product of appropriate rules on each of $\tilde{T}$ and $\tilde{T}'$. For these we have Graham, Hackbusch & Sauter (1997) :

Lemma 2 If we employ a quadrature rule with degree of precision 2 (i.e. it is exact for quadratic functions) on each of $\tilde{T}, \tilde{T}'$ then for each constant $\delta > 0$, (5) is ensured for all p, q satisfying

$$|\boldsymbol{x}_p - \boldsymbol{x}_q| \ge \delta \tag{7}$$

as $h \to 0$. $\square$

For each δ fixed and h sufficiently small, there will exist pairs of nodes $\boldsymbol{x}_p$, $\boldsymbol{x}_q$ which do not satisfy (7) but for which none of the integrals in (4) are singular. For these (5) can only be ensured by requiring the degree of precision of the quadrature rule to increase with h - a logarithmic increase will suffice. However since these results are *asymptotic* as $h \to 0$, it is not clear, for any particular h whether it should be necessary to use rules of precision > 2 for the nonsingular integrals on nearby triangles. In this paper we give numerical results using rules of precision 2 for all the non-singular integrals in (4) and we optain optimal rates of convergence (at least in the L_2 norm) for moderately fine h.

So why is the discrete Galerkin method still expensive? Suppose K_{pq} in (4) is assembled by cycling through all pairs of triangles T, T', computing (6) and then adding the relevant terms in (4) together as they are computed. Recalling the remarks above, we note that it may be useful to use precision 2 rules which are part of a family of higher-order rules, so that precision could be increased when required. One such family on triangles consists of the conical Gauss rules, see e.g. Stroud (1973). For these, precision 2 requires 4 evaluations on each triangle, so that the integration of (6) requires 16 evaluations of the kernel $k(\boldsymbol{x}, \boldsymbol{y})$. (We use the number of kernel evaluations as a measure of the cost of evaluating (4).) If then n_T is the number of triangles in the mesh, the overall cost of doing all the integrals (6) is at least $16n_T^2 \approx 64n^2$ for large n. Here we have used the well-known fact that $n_T \approx 2n$ and have made the somewhat optimistic assumption that all the integrals (6) can be done by 16 kernel evaluations. In reality the cost may be somewhat more than this. If we use the precision 2 rule with nodes at mid-points of the triangle sides, the $64n^2$ becomes $36n^2$. More reductions are possible only at the expense of vastly increased storage requirements. The hybrid method which we now describe would require (under the assumptions above) only n^2 kernel evaluations for the computation of n^2 entries of the matrix K (ignoring, as above, the singular integrals).

The Hybrid Boundary Element Method

The starting point for the hybrid method is the following idea (which may be of interest in other contexts besides accelerating the Galerkin matrix assembly). Consider the integral

$$\int_\Gamma F(\boldsymbol{x})\phi_p(\boldsymbol{x})dx \ , \tag{8}$$

with F a smooth function on $\mathrm{supp}(\phi_p)$. Using the mapping $\boldsymbol{\eta} : \tilde{\Gamma} \to \Gamma$, this may be written

$$\int_{\tilde{\Gamma}} (F \circ \boldsymbol{\eta})(\tilde{\boldsymbol{x}})g(\tilde{\boldsymbol{x}})\tilde{\phi}_p(\tilde{\boldsymbol{x}})d\tilde{x} =: \int_{\tilde{\Gamma}} \tilde{F}(\tilde{\boldsymbol{x}})\tilde{\phi}_p(\tilde{\boldsymbol{x}})d\tilde{x}. \tag{9}$$

If $\mathrm{supp}(\tilde{\phi}_p)$ lies entirely within one of the charts $\tilde{C}_i$, then $\tilde{F}$ is a smooth function on $\mathrm{supp}(\tilde{\phi}_p)$. The function $\tilde{\phi}_p$ is not smooth, but instead of the usual remedy of breaking (9) up into a sum of integrals over each $\tilde{T} \in \mathrm{supp}(\tilde{\phi}_p)$, we treat $\tilde{\phi}_p$ as a weight function and propose the rule:

$$\int_{\tilde{\Gamma}} \tilde{F}(\tilde{\boldsymbol{x}})\tilde{\phi}_p(\tilde{\boldsymbol{x}})d\tilde{x} \approx \sum_l w_{pl} \ \tilde{F}(\tilde{\boldsymbol{x}}_{pl}) \tag{10}$$

In the hybrid method the $\tilde{\boldsymbol{x}}_{pl}$ are selected from the **mesh points** near $\tilde{\boldsymbol{x}}_p$, and w_{pl} are weights which depend on the choice of these points. We can then seek points and weights to ensure that (10) has any desired precision. To ensure precision 2, for example, would require the selection of at least 6 points near $\tilde{\boldsymbol{x}}_p$. If $n_p \geq 6$ points are chosen, then the requirement that (10) has precision 2 leads to the solution of a $6 \times n_p$ system for the weights $\{w_{pl}\}$. This must be done for each p - an $O(n)$ operation globally.

A key question is whether the resulting quadrature rules are *stable* in the sense that

$$\frac{\sum_l |w_{pl}|}{\sum_l w_{pl}} \leq C \tag{11}$$

with C independent of p and h, as $h \to 0$. Rules of precision 2 will be unstable if a quadratic bivariate function happens to (nearly) pass through the chosen quadrature points. In practice this rarely happens and in Graham, Hackbusch & Sauter (1997) we shall present very general criteria for stability which give ways of choosing 6 stable points near any $\tilde{\boldsymbol{x}}_p$ covering most practical situations. These criteria do not assume any uniformity or regularity properties of any particular mesh. We also emphasise that there is great freedom in choosing the $\tilde{\boldsymbol{x}}_{pl}$ - note, for example, that there is no requirement that the points $\tilde{\boldsymbol{x}}_{pl}$ should lie in the support of $\tilde{\phi}_p$.

In practice it is often convenient to choose more than 6 points $\tilde{\boldsymbol{x}}_{pl}$ for each p. In a uniform rectangular mesh, each $\tilde{\boldsymbol{x}}_p$ is connected to 6 others as well as itself, making 7 an obvious number in this case. In fact algorithmically it may be useful to choose, for any p, the node $\boldsymbol{x}_p$ together with all nodes connected to it by edges of the mesh, provided there are at least 6 of these. If $n_p > 6$ then the resulting system to be solved for the weights is overdetermined. It can be solved to obtain the minimum $2-$ norm solution using the QR factorisation - see Golub & van Loan (1989, Section 5.7). We also show in Graham, Hackbusch & Sauter (1997) that if any 6 of the chosen $n_p \geq 6$ nodes yield a stable quadrature rule, then all the n_p points are also stable, provided the underdetermined equations are solved by the QR method.

The approach we have just described yields methods for computing (9) which require only evaluations of F at nodes of the mesh **and not at any other points**. In the illustration above we assumed that supp$(\tilde{\phi}_p)$ lay entirely within one chart $\tilde{C}_i$. If parts of supp$(\tilde{\phi}_p)$ are in different charts we simply separate (9) into a sum of contributions over each relevant chart and repeat the above arguments, where now weights have to be computed for more than one quadrature rule and the corresponding weight function in each case is only part of the function $\tilde{\phi}_p$. In any case we get again a method for computing (9) using only nodal evaluations of F in the form (10).

Now let us return to the question of computing the Galerkin matrix entries K_{pq} given by (3). Again employing $\boldsymbol{\eta} : \tilde{\Gamma} \to \Gamma$ in (3), we can write

$$K_{pq} = \int_{\tilde{\Gamma}} \tilde{\phi}_p(\tilde{\boldsymbol{x}}) \int_{\tilde{\Gamma}} H(\tilde{\boldsymbol{x}}, \tilde{\boldsymbol{y}}) \tilde{\phi}_q(\tilde{\boldsymbol{y}}) d\tilde{y} d\tilde{x} \ , \tag{12}$$

where $H(\tilde{\boldsymbol{x}}, \tilde{\boldsymbol{y}}) := g(\tilde{\boldsymbol{x}})k(\boldsymbol{\eta}(\tilde{\boldsymbol{x}}), \boldsymbol{\eta}(\tilde{\boldsymbol{y}}))g(\tilde{\boldsymbol{y}})$. Using the formula (10) for both p and q we have the approximation

$$\tilde{K}_{pq} = \sum_{l} \sum_{l'} w_{pl} H(\tilde{\boldsymbol{x}}_{pl}, \tilde{\boldsymbol{x}}_{ql'}) w_{ql'} \ . \tag{13}$$

In Graham, Hackbusch & Sauter (1997) we have the following result:

Theorem 3 Suppose (10) has degree of precision 2 and remains stable as $h \to 0$. Suppose also that the meshes are refined in a quasi-uniform way. If we replace K_{pq} by $\tilde{K}_{pq}$ for all p, q with $|\boldsymbol{x}_p - \boldsymbol{x}_q| \geq \delta$, ith $\delta > 0$ independent of h, then the resulting "hybrid" solution $\tilde{u}_h$ satisfies the same accuracy estimate as the Galerkin method itself: $\|u - \tilde{u}_h\|_{-1/2} \leq Ch^{5/2}$, as $h \to 0$, for sufficiently smooth u.

Remark 1 Once the $O(n^2)$ elements of the "Nyström" matrix $(k(\boldsymbol{x}_p, \boldsymbol{x}_q))_{p \neq q}$ have been computed, $\tilde{K}_{pq}$ in (13) can be computed (in the nonsingular case) without further kernel evaluations.

Remark 2 The computation of the weights $\{w_{pl}\}$ and the surface area transformations $g(\tilde{\boldsymbol{x}}_p)$ for each p is an $O(n)$ operation.

Remark 3 The formula (13) computes a sufficiently accurate approximation to the Galerkin matrix entry K_{pq} away from the diagonal by local averaging of the Nyström matrix entries. On and near the diagonal we use the true Galerkin method. This is a *hybrid Nyström-Galerkin* method. When k in (1) is symmetric, so is the Galerkin matrix K, and the hybrid approximation $\tilde{K}$ retains this symmetry.

Remark 4 A *hybrid collocation-Galerkin* method can also be obtained by using the quadrature rule (10) only on the outer integral of (12). However this is of less practical interest.

Remark 5. The stability of the hybrid quadrature rule has only been established in Graham, Hackbusch & Sauter (1997) for the precision 2 case. There is nothing to prevent the practitioner using the rule for higher precision calculations. Our experience is that such rules rarely become unstable and it would be easy to avoid instability in practice if it ever arose by only accepting quadrature rules for which the ratio in (11) is below some a priori upper bound.

Numerical Experiments

In this section we report on some numerical experiments with the Galerkin and the hybrid method for the numerical solution of the single layer potential equation (1), where Γ is the unit sphere $\{\boldsymbol{x} \in \mathbb{R}^3 : \|\boldsymbol{x}\| = 1\}$. The surface Γ is described using 8 charts composed of the triangular faces of a double pyramid $\tilde{\Gamma}$ with nodes $(\pm 1, \pm 1, \pm 1)$ inscribed in Γ. The mapping $\boldsymbol{\eta} : \tilde{\Gamma} \to \Gamma$ is given (globally in this case) by $\boldsymbol{\eta}(\boldsymbol{x}) = \boldsymbol{x}/|\boldsymbol{x}|$.

Meshes are constructed on Γ by starting with the triangles comprising $\tilde{\Gamma}$ itself (Level 1) and performing 4 levels of hierarchical refinement, with each new level obtained from the previous by introducing new nodes at the mid-points of sides and subdividing each triangle into four. This yields a sequence of five meshes on $\tilde{\Gamma}$ with 6, 18, 66, 258 and 1026 nodes, respectively.

Since Γ is the unit sphere it is easy to show (using Green's identity) that

$$\mathcal{K}\left(\frac{\partial u}{\partial n} + \frac{1}{2}u\right) = 2\pi u$$

when u is any harmonic function on Ω. This leads to a class of useful test examples for numerical methods for (1) in which, for special given f, the exact solution is known. In the following experiments we focus on two problems with exact solution:

- **Problem 1:** $u(\boldsymbol{x}) = (3/2)/|\boldsymbol{x} - (2,0,0)|^3$.
- **Problem 2:** $u(\boldsymbol{x}) = (5/2)(2x_1^2 - x_2^2 - x_3^2)$.

In the experiments below u^1 and u^2 denote the true solutions to Problems 1 and 2, respectively. In all the experiments given here we shall report errors in the L_2 norm on Γ, denoted $\|\cdot\|_2$. The energy norm for this problem is $H^{-1/2}$, as described above. In Graham, Hackbusch & Sauter (1997) we shall report convergence rates in this norm. Here we merely note that (assuming stability) the optimal rate of convergence in L_2 is $O(h^2)$ and this is the rate which we should be looking for in practice. In all the tables presented below, columns headed "EOC" contain estimated orders of convergence for the numbers in the column immediately on the left.

Our first set of results is for the Galerkin method itself, Table 1. Here, for the singular integrals, we used the transformation procedures described in Sauter (1996), which reduce these integrals to the sums of regular integrals over 4-dimensional regions consisting of products of $[0,1]^d$ with the simplex of diameter h in $\mathbb{R}^{4-d}$. For the integrals over $[0,1]^d$ we employed tensor products of the 4-point Gauss rule. For the integrals over the simplices we employed conical Gauss rules with 2 points in each coordinate direction. For all the nonsingular integrals (6) we employed conical Gauss rules with 4 points in each triangle. In Table 1, u_h^1 and u_h^2 denote the Galerkin solutions to Problems 1 and 2, respectively. The results show convergence rates approaching $O(h^2)$ for each of these problems.

$Levels$	$\|u^1 - u_h^1\|_2$	EOC	$\|u^2 - u_h^2\|_2$	EOC
1	1.03		3.00	
		2.9		0.4
2	1.35(-1)		2.39	
		1.8		2.8
3	3.84(-2)		3.49(-1)	
		1.9		2.3
4	1.02(-2)		6.92(-2)	
		1.9		1.9
5	2.82(-3)		1.89(-2)	

Table 1: Galerkin Method

Tables 2 3 and 4 show our first results for the hybrid Nyström - Galerkin method, with approximate solutions denoted $\tilde{u}_h^1$, $\tilde{u}_h^2$. Here we have used the simplest possible version of this method, namely we only use the hybrid approximation (13) when $\tilde{\boldsymbol{x}}_p$ and $\tilde{\boldsymbol{x}}_q$ are both interior nodes of charts and when p and q satisfy (7) for various δ. In this case $\tilde{\boldsymbol{x}}_p$ and $\tilde{\boldsymbol{x}}_q$ are both connected to 6 other nodes of the mesh. For each p the rule (10) has 7 points and the weights w_{pl} are computed by the QR method mentioned above. Thus the hybrid method is not used for all the entries which it could be, but nevertheless as the number of levels increases a more substantial proportion of the entries is done this way. In the following tables δ is the largest number such that (7) holds for all p, q for which the hybrid approximation is made and % denotes the percentage of matrix entries which are computed using the hybrid approximation.

$Levels$	δ	%	$\|u^1 - \tilde{u}_h^1\|_2$	EOC	$\|u^2 - \tilde{u}_h^2\|_2$	EOC
3	1.15	9.4	3.76(-2)		3.77 (-1)	
				1.3		2.0
4	0.46	39.0	1.54(-2)		9.48(-2)	
				1.1		1.1
5	0.20	65.7	7.24(-3)		4.34(-2)	

Table 2: Hybrid Method with $\delta = O(h)$

Our first finding is that the precision 2 hybrid method is not as effective near the diagonal of the matrix as the precision 2 discrete Galerkin method. In Table 2 we give the results when the hybrid approximation is used for all p, q for which $K_{p,q}$ does not contain any singular integrals, and for which $\tilde{\boldsymbol{x}}_p$, $\tilde{\boldsymbol{x}}_q$ are interior nodes of charts. That means that (7) is true with $\delta = O(h)$ as the number of levels increases. We see that the rate of convergence of this hybrid method is only about $O(h)$. (It may be argued that the method is still quite effective since the actual Level 5 error for the very much cheaper hybrid method is only slightly more than twice the corresponding error for the Galerkin method.) Note that since the hybrid approximation has no effect, unless there exist interior nodes of charts, the hybrid results start at Level 3.

Table 3 on the other hand gives results where (7) is satisfied for fixed $\delta \in [1.42, 1.53]$. Here the percentage of the matrix which is done by the hybrid method is considerably smaller but it still represents a considerable saving in the Level 5 case. The actual errors are very close to those for the Galerkin method.

Levels	δ	%	$\|u^1 - \tilde{u}_h^1\|_2$	EOC	$\|u^2 - \tilde{u}_h^2\|_2$	EOC
3	1.52	6.6	3.90 (-2)		3.78 (-1)	
				1.8		2.4
4	1.43	19	1.09(-2)		6.96(-2)	
				1.8		1.9
5	1.42	28	3.20(-3)		1.90(-2)	

Table 3: Hybrid Method with $\delta \in [1.42, 1.53]$

Levels	δ	%	$\|u^1 - \tilde{u}_h^1\|_2$	EOC	$\|u^2 - \tilde{u}_h^2\|_2$	EOC
3	1.15	9.4	3.76(-2)		3.77 (-1)	
				1.7		2.4
4	1.24	23.4	1.11(-2)		6.96(-2)	
				1.6		1.8
5	1.12	39.4	3.56(-3)		1.98(-2)	

Table 4: Hybrid Method with $\delta \in [1.12, 1.24]$

Table 4 represents an intermediate situation in which the value of δ is bounded below but at a smaller level than in Table 3. In fact the value of δ at Levels = 3 corresponds to applying the hybrid method to all p, q such that (4) contains only non-singular integrals.

Finally to give some indication of the likely significance of the above results in actual computation times, we observe that our code, which is implemented in MATLAB, takes 16 flops for a single kernel evaluation (recall $k(\boldsymbol{x}, \boldsymbol{y}) = 1/|\boldsymbol{x} - \boldsymbol{y}|$). So, let us consider the most accurate hybrid method: Level 5 of Table 3. Here the number of nodes is $n = 1026$. If we use the standard Galerkin method with the precision 2 conical Gauss rule then (according to the complexity analysis above) this requires about 64 kernel evaluations for each entry of the matrix. The computation of 28% of the off-diagonal matrix entries using this method, taking account of symmetry, is $(14/100) \times ((1026)^2) \times (64) \times (16) \approx 1.5 \times 10^8$ flops. However the cost is about 64 times less if these are done by the hybrid method: about 2.3×10^6 flops. The cost of an LU factorisation for a matrix of size 1026×1026 (in this case using the MATLAB "\" command) is 7.2×10^8. So we see that the hybrid method in its most accurate form produces a saving in the far field matrix computation which is of the same order of magnitude as the cost of the linear solve. Further details will be in Graham, Hackbusch & Sauter (1997).

Acknowledgement This work was supported by British Council/DAAD ARC Grant Number 590 and by a Visiting Fellowship for I.G. Graham from the Leibniz Prize Fund held by Professor W. Hackbusch at the Christian Albrechts University of Kiel.

References

Erichsen, S. and Sauter, S.A. (1996) Efficient automatic quadrature in 3-d Galerkin BEM. To appear in *Comp. Meth. Appl. Mech. Engrg.*

Golub, G.H. and Van Loan, C.F. (1989) *Matrix Computations.* Johns Hopkins University Press, London, 1989.

Graham, I.G. , Hackbusch, W. and Sauter, S.A. (1997) Hybrid Galerkin Boundary Elements: Theory and Implementation. *In preparation.*

Hackbusch, W. and Sauter, S.A. (1993) On the efficient use of the Galerkin method to solve Fredholm integral equations. *Applications of Mathematics*, **38**, 301-322.

Sauter, S.A. and Krapp, A. (1996) On the effect of numerical integration in the Galerkin boundary element method. *Numerische Mathematik*, **74**, 337-359.

Sauter, S.A. (1996) Cubature techniques for 3-D Galerkin BEM, in *Boundary Elements: Implementation and Analysis of Advanced Algorithms*, pp. 29-44, W. Hackbusch and G. Wittum, Eds, Notes in Numerical Fluid Mechanics 54, Vieweg, Braunschweig.

Stroud, A.H. (1973) *Approximate Calculation of Multiple Integrals.* Prentice-Hall, Englewood Cliffs, N.J.