On the Efficient Realization of Sparse Matrix Techniques for Integral Equations with Focus on Panel Clustering, Cubature and Software Design Aspects

W. Hackbusch, C. Lage, and S.A. Sauter

Institut für Informatik und Praktische Mathematik,
Christian-Albrechts-Universität Kiel,
D-24098 Kiel, Germany

Summary. The method of integral equations is an elegant tool where boundary value problems on a domain Ω are transformed into integral equations defined on the surface of Ω. The discretization via the boundary element method (BEM) has several advantages compared to FE-discretizations of PDEs on the whole domain Ω. On the other hand, the most significant drawback of the BEM is that the system matrix is full and, in addition, the computation of the elements requires the evaluation of complicated surface integrals. In this paper we show how to avoid the generation of the whole system matrix by means of the *panel clustering method* which represents the discrete operator in an alternative form. Only matrix elements close to the diagonal have to be computed. Furthermore, we will present new semi-analytic techniques for computing those *nearfield matrix entries* efficiently.

From the view point of software design aspects the efficient realization of boundary elements, especially the panel clustering algorithm, is a non-trivial task. We will explain how the complexity of implementing the multifarious BEM can be managed by object-oriented design methods.

1. Introduction

The boundary element method (BEM) is an elegant tool for solving elliptic boundary value problems numerically. First, the method of integral equations is applied transforming the PDE on a domain Ω into an integral equation defined on the boundary of Ω. This integral equation can be discretized by Petrov-Galerkin methods defined on the surface of Ω. Instead of the discretization of the whole domain, only the lower dimensional boundary of Ω has to be partitioned into a FE-grid. This is one of the major advantages of the BEM. Especially for 3-d problems, grid generation of the whole domain Ω in many cases is still an extremely time consuming step. As a consequence, the dimension of the stiffness matrix is much smaller as for the corresponding FE-discretization. Furthermore, the matrix condition number is smaller compared to the FEM-system governing the convergence speed of iterative solvers applied to the linear system. On the other hand, the major drawback of the BEM is that the arising system matrix is full and, in addition, the computation of the matrix entries requires the evaluation of complicated surface integrals. Recently, many attempts have been made to overcome these two

difficulties. The full matrix can avoided by representing the integral operator on the discrete level in an alternative form, which is based on the approximation of the kernel function of the integral operator. The panel clustering, originally developed by Hackbusch and Nowak (see [8]) for the collocation discretization of the Laplace equation, turned out to be also applicable to any Petrov-Galerkin method, where we emphasize the important case of the Galerkin method where test and trial space coincide. A related method is the multipole expansion of Rokhlin (see [14]) which, however, can be applied only to Nyström discretizations and not, e.g., to the important case of the hypersingular formulation of integral equations. A further approach in this direction was presented by Brandt and Lubrecht in [4], [5] where multi-grid techniques together with high order inter-grid interpolation is used to condensate the system matrix.

Another method to condensate the system matrix is the application of wavelets to boundary integral equations. It turns out that, by using a suitable wavelet basis, the off-diagonal entries of the system matrix have a strongly decreasing behaviour with increasing distance from the diagonal (see [6], [19]).

All condensation strategies have in common that the *nearfield matrix entries*, i.e., those elements lying close to the diagonal have to be computed in the standard matrix-oriented way. As mentioned above, this is a non-trivial and time-consuming task in BEM computations. The product of two basis functions with a possibly very complicated *kernel function* having a singular or nearly singular behaviour has to be integrated over surface pieces. Due to the singular behaviour of the kernel function, standard cubature techniques lose their accuracy. We will show that, for kernel functions arising by discretizing boundary value problems via the BEM, the application of suitable coordinate transforms enables us to integrate the variable containing the singularity analytically resulting in an integrand which has no or significantly reduced singular behaviour. For hypersingular integrals, we write the arising regularization in a form which can be evaluated in many cases analytically resulting in surprisingly easy formulae.

In this paper, we discuss thoroughly the software design aspects which came up by realization of the developed numerical algorithms in a computer code. New data structures have to be developed such that, e.g., the panel clustering technique (where non-standard quantities as the *farfield coefficients*, the *expansion coefficients*, and the *clusters* have to be organized efficiently) performs optimally. Also emphasize is taken on parallelization strategies.

The paper is organized as follows. In the next chapter, we go briefly through the setting of Petrov-Galerkin discretization of boundary integral equations and introduce the notations necessary for the sequel. Then, in Chapter 3., we investigate the computation of the surface integrals arising when computing the matrix entries of the linear system. First, we have to provide some analytical properties of the arising kernel functions which then will be used in the design of efficient cubature strategies. (Note that in more

than one dimension the terminus *quadrature* is replaced by *cubature*). In Chapter 4., we explain the panel clustering algorithm. After having developed the basic principle, the algorithm is formulated and an error analysis is presented. Estimates of the asymptotic complexity of the algorithm follows and remarks on their relevance for practical problem sizes are given. Chapter 6. is devoted to the design of modern BEM software in C++. It will be explained which data structures are well suited in order to make the different algorithms efficient and flexible with respect to different kinds of integral equations, discretization schemes as, e.g., collocation or Galerkin BEM, geometries and cubature techniques. In the last chapter numerical experiments are reported which show the performance of the panel clustering algorithm for different discretizations and formulations of integral equations.

2. Setting and Preliminaries

Let us consider an elliptic boundary value problem on a domain $\Omega \subset \mathbb{R}^d$ of order $2m$ of the form

$$\begin{aligned} Du &= 0, \qquad \text{in } \Omega \\ Bu &= r, \qquad \text{on } \Gamma := \partial\Omega \end{aligned} \tag{2.1}$$

with a system of m boundary differential operators B_j having mutually different order. If Ω is unbounded, suitable radiation conditions have to be imposed. We assume that the operator D has constant coefficients and the fundamental solution defined by

$$\begin{aligned} &DS = \delta_0, \qquad \text{in } \mathbb{R}^3, \\ &S \text{ satisfies the radiation conditions,} \end{aligned} \tag{2.2}$$

is known explicitly. Here and in the following, δ_x denote the Dirac-functional located at the point x. In order to exhibit the principal ideas, it is sufficient to make the following assumptions, while we state that all presented methods can be applied to the general situation without any significant modification.

- the space dimension d equals 3,
- the order $2m$ of the operator is 2,
- $B = (\partial/\partial n)^j$ with j is either 0 or 1, whereas $\partial/\partial n$ denotes the normal derivative,
- the surface Γ is Lipschitz continuous.

Using the (direct) method of integral equations, problem (2.1) is transformed into an integral equation which, in the variational formulation, is given by seeking $u \in X$ such that

$$\langle \lambda u, v \rangle + \langle K_1 u, v \rangle = \langle K_2 r, v \rangle \,, \qquad \forall v \in Y \tag{2.3}$$

is satisfied. Here, X, Y are suitable function spaces and $\langle \cdot, \cdot \rangle$ is a dual pairing. The operators K_1 and K_2 denote integral operators which will be specified later. The function λ is piecewise constant on Γ. The reason for presenting the formulation in this abstract setting is that most of the theory and algorithms discussed in the sequel can be presented in a more compact form as if discussed for each choice of the spaces and pairing above separately. The Petrov-Galerkin method is characterized by defining finite dimensional subspaces X_n and Y_n of X and Y, i.e, finding $u_n \in X_n$ such that (2.3) is satisfied for all $v_n \in Y_n$. The definition of the discrete spaces is typically based on a geometrical partitioning of the surface Γ. For this, let $T_N := \{\Delta_j : 1 \leq j \leq N\}$ be a partitioning of Γ into small surface pieces. Here, we assume that T_N is a triangulation in the sense that there exists a family of diffeomorphisms $\{\chi_i\}_{1 \leq i \leq N}$ mapping the *master element*, i.e., the triangle with vertices $(0,0)^T$,$(1,0)^T$, and $(1,1)^T$, onto the elements $\{\Delta_i\}_{1 \leq i \leq N}$. We assume that T_N satisfies the following conditions:

1. $\Gamma = \bigcup_{\Delta \in T_N} \Delta$,
2. $\Delta_i \cap \Delta_j$ is either empty, a vertex, a (curved) edge or a (curved) triangle,
3. $h := \max\limits_{\Delta \in T_N} \operatorname{diam}(\Delta) \leq C \operatorname{diam}(\Delta), \qquad \forall \Delta \in T_N,$
4. $\max\limits_{B \text{ is a ball contained in } \Delta} \operatorname{diam}(B) \geq Ch, \qquad \forall \Delta \in T_N.$

The finite element space X_n is given by lifting polynomials of degree p defined on the master element onto the surface, i.e.,

$$X_n := \left\{ u \in \mathcal{C}^k(\Gamma) : u \circ \chi_i \mid_{\hat{\Delta}} \text{ is a polynomial of degree } p \text{ for all } 1 \leq i \leq N \right\}.$$

Let $\Theta_n = \{x_i : 1 \leq i \leq n\} \subset \Gamma$ denote the set of uni-solvent nodal points having the property that the interpolation problem of finding $u \in X_n$ such that $u(x_i) = f_i$ for all $1 \leq i \leq n$ has a unique solution for any given vector $f \in \mathbb{C}^n$. The example below illustrates typical choices of discrete spaces X_n, Y_n and dual pairings.

Example 2.1. 1. The Collocation Method is characterized by the following choices. X_n is defined as explained above and $Y_n := \operatorname{span}\{\delta_x : x \in \Theta_n\}$. The dual pairing is given by $\langle w, \delta_x \rangle = w(x)$. Explicitly, (2.3) takes the form: find $u \in X_n$ such that

$$\lambda(x)\, u(x) + K_1[u](x) = K_2[r](x), \qquad \forall x \in \Theta_n. \tag{2.4}$$

2. The Galerkin Method is characterized by choosing X_n as above, $Y_n = X_n$ and $\langle w, u \rangle = (w, u)_0$, where $(\cdot, \cdot)_0$ denote the L^2-scalar product on the surface Γ. This results in seeking $u \in X_n$ such that

$$(\lambda u, v)_0 + (K_1 u, v)_0 = (K_2 r, v)_0, \qquad \forall v \in X_n. \tag{2.5}$$

In the following we collect some properties of the integral operators $K_{1,2}$ which will be used in the next chapters. All integral operators appearing in the context of solving elliptic boundary value problem via the method of integral equations can be written in the form

$$K\left[u\right](x) = \text{p.f.} \int_{\Gamma} k\left(x, y, y-x\right) u\left(y\right) dy. \tag{2.6}$$

Some explanations are necessary. The kernel function $k\left(x, y, y-x\right)$ is a suitable Gâteau derivative (of order less or equal than $2m$) of the fundamental solution defined by (2.2). The fundamental solution behaves singularly only for $x = 0$ and is smooth elsewhere. In the three-dimensional case, we know (cf. [11]) that it can be estimated as

$$\left|S\left(x-y\right)\right| \leq C \left\|x-y\right\|^{2m-3}, \qquad \forall x, y \in \Gamma \text{ with } x \neq y.$$

Hence, it is natural to assume that

$$\left|k\left(x, y, y-x\right)\right| \leq C \left\|x-y\right\|^{-s}, \qquad \forall x, y \in \Gamma \text{ with } x \neq y, \tag{2.7}$$

where the *order of the singularity* s is an integer smaller or equal than 3. With respect to the first both variables, k is smooth in smooth parts of the surface Γ and may jump across edges and corners. An important case is the case of flat patches of the surface. In such regions, it is natural to assume that k is constant in the first two variables, i.e.,

$$k\left(x, y, y-x\right) = k\left(x-y\right). \tag{2.8}$$

For $s = 2, 3$ the kernel function is not Lebesgue-integrable. We have to apply the concept of regularized integrals in the sense of Hadamard. Let $B_{\epsilon}\left(x\right)$ denote a ball with radius ϵ about x and γ be a surface piece of Γ. Then, the functional J is well-defined by

$$J_{\gamma}\left[u\right](\epsilon, x) := \int_{\gamma \setminus B_{\epsilon}(x)} k\left(x, y, y-x\right) u\left(y\right) dy.$$

It was shown in [21] that, for all kernels which arise by transforming elliptic boundary value problems into integral equations, the function J admits an expansion w.r.t ϵ of the form

$$J_{\gamma}\left[u\right](\epsilon, x) = A_{-1}\left(x\right) \epsilon^{-1} + A_{\log}\left(x\right) \log \epsilon + A_{0}\left(x\right) + o\left(1\right).$$

The *finite part* of such an expansion is given by

$$\underset{\epsilon}{\text{p.f.}}\, J_{\gamma}\left[u\right](\epsilon, x) := A_{0}\left(x\right).$$

In order to exhibit the dependency of A_0 on γ and u we alternatively write $A_{\gamma}\left[u\right]$. We define the regularized integral (2.6) by

$$\text{p.f.} \int_\gamma k(x, y, y-x)\, u(y)\, dy := \operatorname*{p.f.}_{\epsilon} J_\gamma[u](\epsilon, x) = A_0(x). \tag{2.9}$$

Now, all general assumptions are collected which will be needed for discussing cubature methods for the arising integrals and approximation of the kernel functions by means of the panel clustering method.

3. Cubature Techniques for the Approximation of Singular and Nearly Singular Surface Integrals Arising in BEM

In this chapter, we present cubature techniques for collocation and Galerkin discretizations of boundary integral equations. We start with the abstract Petrov-Galerkin formulation. Let $\{\phi_i\}_{1\le i\le n}$ and $\{\varphi_i\}_{1\le i\le n}$ denote a basis of X_n and Y_n respectively. Any function $u \in X_n$ is uniquely linked to a coefficient vector $u \in \mathbb{C}^n$ by the basis representation

$$u(x) = \sum_{i=1}^{n} u_i \phi_i(x). \tag{3.1}$$

The Petrov-Galerkin discretization is then equivalent to solving the linear system,

$$(\boldsymbol{M} + \boldsymbol{K})\, u = f$$

with $n \times n$ matrices $\boldsymbol{M}$ and $\boldsymbol{K}$ and the n-vector f defined by

$$\begin{aligned} \boldsymbol{M}_{i,j} &:= \langle \lambda\phi_j, \varphi_i \rangle, \\ \boldsymbol{K}_{i,j} &:= \langle K_1\phi_j, \varphi_i \rangle, \\ \boldsymbol{f}_i &:= \langle K_2 r, \varphi_i \rangle. \end{aligned}$$

The computation of the *mass matrix* $\boldsymbol{M}$ is trivial, since no integral operator is involved and $\boldsymbol{M}$ usually is sparse. The computation of $\boldsymbol{f}$ is similar as the computation of $\boldsymbol{K}$, thus, we restrict our attention on the approximation of the entries of $\boldsymbol{K}$. In the following we write K instead of K_1. In the rest of this paper we assume that

1. the integral operator K arises by transforming an elliptic boundary value problem into an integral equation,
2. the trial space X_n is a subspace of the (continuous) function space X,
3. $X_n \subset \mathcal{C}^{0,1}(\Gamma)$ for hypersingular equations, i.e., $s = 3$.

An important property of the application of K to basis functions $\phi_i \in X_n$ is stated in the following

Theorem 3.1. *Let γ be a simple connected surface piece which consists of some triangles of T_N. Recall the definition of the part-fini integral (see (2.9)) and the order s of the singularity of the kernel function (see (2.7)). Let*

$$A_\gamma[\phi_i](x) := \text{p.f.} \int_\gamma k(x, y, y-x)\, u(y)\, dy.$$

Then,

$$|A_\gamma[\phi_i](x)| \leq \begin{cases} C & \text{if } s \leq 1, \\ C \log \operatorname{dist}(x, \partial\gamma) & \text{if } s = 2, \\ C \operatorname{dist}^{-1}(x, \partial\gamma) & \text{if } s = 3, \end{cases} \tag{3.2}$$

with C independent of x.

For $\gamma = \Gamma$ and any $s \leq 3$,

$$|A_\Gamma[\phi_i](x)| \leq C, \qquad \forall x \in \Gamma$$

is satisfied.

Proof. The proof of this theorem can be found in [12], [21], or [13].

In the following subsection, we consider cubature techniques for the collocation method.

3.1 Cubature Techniques for the Collocation Method

From Theorem 3.1, it follows that it is problematic to use the collocation methods for strongly singular integral equations, i.e., $s \geq 2$, since, typically, one would have to evaluate $A_\gamma[\phi_i]$ at corner and edge points. However, estimate (3.2) does not necessarily imply that, for $x \in \partial\gamma$, the function $A_\gamma[u](x)$ is infinite. Typically for Cauchy-singular integral equations, i.e., $s = 2$, the function $A_\gamma[u](x)$ tends to infinity as x approaches $\partial\gamma$ but remains finite at points $x \in \gamma$. This behaviour is thoroughly analyzed and correction functionals can be computed (see [21] and [22]). In the cited papers, it was shown that it is always possible to split the integral into some point functionals and an integral having a weakly singular integrand. Then, either Duffy's triangle coordinates or polar coordinates smoothes the integrand such that it can be treated efficiently with properly scaled Gauß-Legendre formulae. Here, we do not go into the details but proceed with considering the nearly singular situation. The arising problem is to compute

$$I_\Delta(x_i) := \int_\Delta k(x_i, y, y - x_i)\, \phi_i(x)\, dx, \tag{3.3}$$

where x_i denote a collocation point having a positive distance from the surface panel Δ,

$$\operatorname{dist}(x_i, \Delta) = \delta > 0.$$

In this case, the part-fini integral reduces to the Rieman integral, since the integrand is regular. On the other hand, the nth derivative of the integrand behaves like $O(n!h^{-n})$ resulting in a very poor convergence speed of Gauß-Legendre formulae. Different attempts have been made to overcome this problem. Subdividing the triangle in the direction of the singularity was presented in [1] and [2]. This method was generalized by introducing variable cubature order in Schwab[20]. We present here a semi-analytical technique which was worked out in detail in [10]. It was shown that the integral (3.3) can be computed to any desired accuracy by computing integrals of the type

$$\int_{\Delta} \frac{(u_1+\delta_{\triangle})^{\nu_1} u_2^{\nu_2} \delta_{\perp}^{\nu_3}}{\sqrt{(u_1+\delta_{\triangle})^2+u_2^2+\delta_{\perp}^2}^{s+t}} d\Delta_u$$

where $\delta_{\triangle}$ denote the horizontal distance of x_i from Δ and $\delta_{\perp}$ the vertical distance. By introducing shifted polar coordinates of the form

$$u = (r-\delta_{\triangle}\cos\alpha)\begin{pmatrix}\cos\alpha\\ \sin\alpha\end{pmatrix}$$

the integral can be written as

$$\delta_{\perp}^{\nu_3}\sum_{i=0}^{\nu_1+\nu_2+1}\int_{\alpha_{\min}}^{\alpha_{\max}}\delta_{\triangle}^{\nu_1+\nu_2+1-i}c_{\nu_1,\nu_2,i}(\sin\alpha,\cos\alpha)\int_{\delta_{\triangle}\cos\alpha}^{R(\alpha)+\delta_{\triangle}\cos\alpha}\frac{r^i}{\sqrt{r^2+\delta_{\triangle}^2\sin^2\alpha+\delta_{\perp}^2}^{s+t}}drd\alpha,$$

where $c_{\nu_1,\nu_2,i}(\cdot,\cdot)$ are polynomials, $R(\alpha)$ is a smooth function and $\alpha_{\min}$, $\alpha_{\max}\in[-\pi,\pi[$. The inner integration can be performed analytically by using

$$\int_a^b \frac{r^n}{\sqrt{r^2+\delta_{\alpha}^2}^s}dr = \frac{1}{\sigma_{n,s}}\frac{p(\delta_{\alpha},r)}{\sqrt{r^2+\delta_{\alpha}^2}^{s-2}}+\kappa_{n,s}\log\left(\pm r+\sqrt{r^2+\delta_{\alpha}^2}\right)\Big|_a^b$$

where p is a polynomial of degree less or equal than $\frac{s+n-3}{2}$ and $\sigma_{n,s}$, $\kappa_{n,s}$ are some coefficients. It can be shown that the integrand of the outer integration has a reduced singular behaviour and can be computed by properly scaled one-dimensional Gauß-Legendre quadrature. A careful error analysis (see [10]) shows that the order of the angular integration has to be increased logarithmically with decreasing distance δ and triangle size diamΔ.

3.2 Cubature Techniques for the Galerkin Method

For the Galerkin discretization of BIE, the elements of the stiffness matrix are given by

$$K_{i,j} := \int_{\operatorname{supp}\phi_i} \phi_i(x) \text{ p.f.} \int_{\operatorname{supp}\phi_j} k(x, y, y-x)\, \phi_j(y)\, dy dx.$$

The support of the basis functions ϕ_i consists of a few triangles. It would be desirable to split the computation of $K_{i,j}$ into a sum of integrals over pairs of panels $\Delta_x \times \Delta_y$. This is not trivial in view of the regularization process involved with the part fini integral. From Theorem 3.1, we know that

$$A_\Gamma[\phi_j](x) := \text{ p.f.} \int_\Gamma k(x, y, y-x)\, \phi_j(y)\, dy dx$$

is bounded and hence,

$$\int_\Delta \phi_i(x)\, A_\Gamma[\phi_j](x)\, dx = \int_\Delta \phi_i(x) \sum_{\Delta_y \in \operatorname{supp}\phi_j} A_{\Delta_y}[\phi_j](x)\, dx.$$

Unfortunately, $A_{\Delta_y}[\phi_j]$ may contain strong singularities at corners and edges of Δ such that the sum may not be interchanged with the outer integration in general. The following concept of introducing an outer regularization process was developed in [13]. For this, let the δ-strip Δ^δ be defined by

$$\Delta^\delta := \{x \in \Delta_x \mid \operatorname{dist}(x, \partial\Omega) \le \delta\}.$$

Then, it was shown in the cited paper that

$$\begin{aligned}
&\int_\Delta \phi_i(x)\, A_\Gamma[\phi_j](x)\, dx \\
&\quad = \sum_{\Delta_y \in \operatorname{supp}\phi_j} \underset{\epsilon}{\text{p.f.}} \underbrace{\int_\Delta \int_{\Delta_y \setminus B_\epsilon(x)} \phi_i(x)\; k(x, y, y-x)\, \phi_j(y)\, dy dx}_{=: K_{\Delta \times \Delta_y}(\epsilon)} \qquad (3.4) \\
&\qquad - \underset{\delta}{\text{p.f.}}\, \underset{\epsilon}{\text{p.f.}} \underbrace{\int_{\Delta^\delta} \int_{\Delta_y \setminus B_\epsilon(x)} \phi_i(x)\; k(x, y, y-x)\, \phi_j(y)\, dy dx}_{=: K_{\Delta \times \Delta_y}(\epsilon, \delta)}.
\end{aligned}$$

First, we consider the approximation of the term $\underset{\delta}{\text{p.f.}}\, \underset{\epsilon}{\text{p.f.}}\, K_{\Delta \times \Delta_y}(\epsilon, \delta)$. In [13], it was shown that this term vanishes for Cauchy-singular and weakly singular integrands and in any case if $\bar{\Delta} \cap \bar{\Delta}_y$ is either empty or a vertex, thus, we may restrict to the case $s = 3$ and $\bar{\Delta} \cap \bar{\Delta}_y$ is either an edge or a panel. Since $\Delta^\delta \to \emptyset$, we may introduce several simplification of the integrand and integration domain which leaves the limit unchanged. It can be shown that Δ^δ may be decomposed into a sum containing integrals over rectangles $e_m \times (0, \delta)$, where e_m denotes an edge of Δ. The triangle Δ_y may be replaced by a suitable square Q_m. Furthermore the basis functions may be replaced by constant continuation of the traces $\phi_i|_{e_m}$ vertical to e_m and the kernel function by its principal part. Finally, the (curved) triangles may be replaced by suitable flat triangles. Thus, it is sufficient to develop techniques to compute

$$I_m^{i,j} := \operatorname*{p.f.}_{\delta} \operatorname*{p.f.}_{\epsilon} \int_{e_m \times (0,\delta)} \int_{Q_m} \tilde{\phi}_{i,m}(x)\; k_{\text{principal}}(y-x)\, \tilde{\phi}_{j,m}(y)\, dydx.$$

We introduce relative coordinates by $z = y - x$ and define the polynomial $b_{i,j}(x,z) := \tilde{\phi}_{i,m}(x)\; \tilde{\phi}_{j,m}(z+x)$, yielding

$$I_m^{i,j} = \int_{D(\delta,\epsilon)} b_{i,j}(x,z)\; k_{\text{principal}}(z)\, dzdx$$

with a suitable four-dimensional domain $D(\delta,\epsilon)$. It turns out that one integration can be carried out analytically due to the polynomial character of $b_{i,j}$. In many cases the remaining integrand can be computed analytically, too. This has be worked out in detail in [13] for the case that the principal part of the kernel function is given by

$$\frac{\partial^2}{\partial n_y \partial n_x} \frac{C}{\|x-y\|},$$

i.e., the underlying differential equation is the Laplace or the Helmholtz equation. The result is that for linear elements and triangles Δ, Δ_y with a common edge

$$I_m^{i,j} := c_{i,j}\, |e_m|\, \frac{2}{9} \left(\frac{\omega}{\tan\omega} + \frac{4}{9} + \frac{1}{3} \log \frac{|e_m|}{2} \right),$$

where ω denote the (smaller) angle between the planes through Δ and Δ_y. The coefficient $c_{i,j}$ is either $0, 1$, or 2 dependent on the indices i, j. This formulae can further be simplified if the case $\Delta = \Delta_y$ is taken into consideration. Here, we skip the details.

Now, we investigate the approximation of the quantity $K_{\Delta\times\Delta_y}(\epsilon)$. Since this integral is regular, we may use the usual rule of transformation of variables. In [9], it was shown how $K_{\Delta\times\Delta_y}(\epsilon)$ can be approximated by a sum of integrals of the same form over flat triangles. Hence, we restrict to this situation and in view of (2.8), we assume that $k(x,y,y-x)$ is constant in the first both variables on the integration domain $D := \Delta \times \Delta_y$. This means that $\tilde{k}(z) := k(x,y,z)$, where we skip the tilde in the sequel. We have to distinguish the following situation.

1. The singular case: $\bar{\Delta} \cap \bar{\Delta}_y \neq \emptyset$.
2. The nearly singular case: $\operatorname{dist}(\Delta, \Delta_y) = \delta > 0$ but "small".
3. The regular case: $\operatorname{dist}(\Delta, \Delta_y) \geq C > 0$ with $C = O(1)$.

These cases will be considered in the following subsections.

3.2.1 The Singular Case: $\bar{\Delta} \cap \bar{\Delta}_y \neq \emptyset$. Due to the space limitations of this paper, we consider here only the case of $\Delta = \Delta_y$, where the general situation is considered in [18], [9] and [13]. The restriction to the case that Δ lies in the (1,2)-plane is not essential but will simplify the notations. We consider the integral

$$Q := \int_{\substack{\Delta \times \Delta \\ \|y-x\| \geq \epsilon}} \phi_i(x)\, k(x-y)\, \phi_j(y)\, dydx.$$

The key point of the transformation procedure is the use of relative coordinates:

$$y = z + x.$$

Let the polynomial b be defined by $b_{i,j}(x,z) := \phi_i(x)\, \phi_j(z+x)$. The integral above then can be written in the form

$$\int_{\substack{\tilde{D} \\ \|z\| \geq \epsilon}} b_{i,j}(x,z)\, k(z)\, dzdx$$

where the domain of integration is a four-dimensional polyhedron given by

$$\tilde{D} := \left\{ (x,z) \in \mathbb{R}^{2\times 2} \mid \exists\, (x,y) \in \Delta \times \Delta : z = y - x \right\}.$$

This polyhedron can be parametrized also in the opposite way such that x depends on z. This means that the integration with respect to z can be interchanged with the x-integration. We obtain

$$Q := \int_{\substack{\tilde{D} \\ \|z\| \geq \epsilon}} k(z)\, b_{i,j}(x,z)\, dxdz.$$

Since b is a polynomial and $\tilde{D}$ is a polyhedron, the x-integration can be carried out analytically yielding a *piecewise* polynomial function $B_{i,j}(z) := \int b_{i,j}(x,z)\, dx$. Splitting the remaining integration domain into subdomains where $B_{i,j}(z)$ is a polynomial and further affine transformations results in

$$Q = \sum_{m=1}^{3} \text{p.f.} \int_{h\|z-A_m\| \geq \epsilon} k(A_m - z)\, H_m^{i,j}(z)\, dz,$$

where the functions $H_m^{i,j}$ are polynomials in z. The remaining integral can be computed using polar coordinates about A_m :

$$Q = \sum_{m=1}^{3} \text{p.f.} \int_{\alpha_{\min,m}}^{\alpha_{\max,m}} \int_{\epsilon/h}^{R(\alpha)} k\begin{pmatrix} r\cos\alpha \\ r\sin\alpha \end{pmatrix} H_m^{i,j}\left(A_m + \begin{pmatrix} r\cos\alpha \\ r\sin\alpha \end{pmatrix} \right) r dr d\alpha.$$

It turns out that the r-integration can be valuated analytically very easily and the limit process, too. The integrand of the remaining angular integration is smooth and can be approximated efficiently using properly scaled Gaussian quadrature rules.

3.2.2 The Nearly Singular Case. In the case that $\operatorname{dist}(\Delta, \Delta_y) =: \delta > 0$ is small the convergence of standard cubature formulae becomes very poor. To overcome this problem, it is possible to write the integral over $\Delta \times \Delta_y$ as a sum over domains containing the singularity and proceeding as explained for the singular case (see [18]). Here, we will explain briefly a further alternative which is very similar as described above for the collocation method (see [13]). Again the starting point is the introduce relative coordinates $y - x = z$ and shifted polar coordinates. The r-integration can be done explicitly resulting in simple formulae for the primitive integrals having a reduced singular behaviour. The remaining three-dimensional integral can be approximated by properly scaled Gauß-Legendre cubature, i.e., the order of integration again has to be increased logarithmically with respect to the diameter of the triangles and the distance δ.

3.2.3 The Regular Case. In this case the kernel function has no singular behaviour. Standard cubature techniques as tensor Gauß rules may be applied having the well known fast convergence properties. However, the question arises how accurate the integrals have to be computed such that the overall asymptotic discretization error is not reduced. In [16] it was shown what degree of exactness the inner integration and the outer integration must have, dependent on the order of the integral operator, on the order of approximation, on distance from the singularity, and the norm in which the error is measured, such that the asymptotic convergence order of the overall discretization is not influenced. In order to illustrate the results, let us assume that linear finite elements are employed, the distance δ from the singularity is $O(1)$ and the integral operator K maps the Sobolev space H^s onto H^{-s}. The error is measured in the H^ρ-norm. The following table shows pairs of numbers, where the first one denotes the degree of exactness of the outer integration and the second one for the inner integration.

s	$\rho = +\frac{1}{2}$	$\rho = 0$	$\rho = -\frac{1}{2}$	$\rho = -1$	$\rho = -\frac{3}{2}$	$\rho = -2$	$\rho = -\frac{5}{2}$	$\rho = -3$
$+\frac{1}{2}$	$(2,1)$	$(1,1)$	$(2,2)$	$(1,2)$	–	–	–	–
0	–	$(2,1)$	$(3,2)$	$(2,2)$	$(3,3)$	$(2,3)$	–	–
$-\frac{1}{2}$	–	–	$(2,2)$	$(3,2)$	$(4,3)$	$(3,3)$	$(3,4)$	$(2,4)$

Table 3.1. Required degree of exactness (γ_x, γ_y) in order to get an optimal discretization order with piecewise linear functions in the H^ρ-norm.

4. The Panel Clustering Method

The boundary element method leads to a system of linear equation which contains a *full* coefficient matrix of dimension n. The reason lies in the fact that

the kernel function $k(x, y, x-y)$ contains a term of the form $\|x-y\|$ linking every point $x \in \Gamma$ with a point $y \in \Gamma$. The panel clustering method was developed by Hackbusch and Nowak in [8], [7] for collocation discretizations and generalized to the Galerkin BEM in [18], [9]. The idea of the method is the approximation of k by a series of the form $k(x, y, y-x) \approx \sum \kappa_{\nu,\mu} \omega_\nu(x) \omega_\mu(y)$ with a suitable function system $\{\omega_\nu\}$. Hence, the x-integration is separated form the y-integration reducing the amount of work substantially. The efficiency of the method depends on how fast the kernel function can be approximated by the function system $\{\omega_\nu\}$, i.e., the number of term in the series and on the question how efficiently the arising quantities can be computed and organized. This will be discussed in the following. We recall here the setting of Petrov-Galerkin methods for boundary integral equations as defined in Chapter 2. and the notation of the trial space by $X_n = \operatorname{span}\{\phi_i\}_{1 \le i \le n}$ and test space $Y_n = \operatorname{span}\{\varphi_i\}_{1 \le i \le n}$.

4.1 Kernel Expansions

The kernel functions appearing in the definition of integral equations are related to the singularity function $s : \mathbb{R}^3 \to \mathbb{R}$ of the underlying boundary value problem. This function can be written in the form

$$s(y-x) = \sum_{|\nu| \ge t} c_\nu \frac{(y-x)^\nu}{\|y-x\|^{s+t}}$$

where $\nu \in \mathbb{N}_0^3$ denote multi-indices and $|\nu| := \nu_1 + \nu_2 + \nu_3$. Let $z_0 := y_0 - x_0$ be "large". Consequently, $s(z) = s(y-x)$ is smooth in a neighborhood $z \in \mathcal{U}(z_0)$. We expand s as a series of order m about z_0:

$$s(z) = \sum_{|\nu|=0}^{m-1} \sum_{\mu=0}^{\nu} \kappa_{\nu,\mu-\nu}(z_0) \omega_\nu(x) \omega_\mu(y) + R_m(z_0, z) \tag{4.1}$$

with some coefficients κ. The kernel function is a suitable Gâteau derivative of s, where we restrict here to normal derivatives. Hence, we can use the expansion

$$\begin{aligned} k(x, y, y-x) \quad = \quad & \sum_{|\nu|=0}^{m-1} \sum_{\mu=0}^{\nu} \kappa_{\nu,\mu-\nu}(z_0) \left(\frac{\partial}{\partial n_x}\right)^{\alpha_1} \omega_\nu(x) \left(\frac{\partial}{\partial n_y}\right)^{\alpha_2} \omega_\mu(y) + \\ & + \frac{\partial^{\alpha_1+\alpha_2}}{\partial n_x^{\alpha_1} \partial n_y^{\alpha_2}} R_m(z_0, y-x). \end{aligned}$$

Example 4.1. Let $s(z) = \|z\|^{-1}$ which is (up to a constant factor) the singularity function of the 3-d Laplace operator. Taylor expansion about $z_0 = y_0 - x_0$ results in

$$s(z) = \sum_{|\nu|=0}^{m-1} \underbrace{\frac{1}{\nu!} \frac{\partial^{|\nu|}}{\partial z^\nu} s(z) \Big|_{z=z_0}}_{=: \kappa_\nu(z_0)} (z - z_0)^\nu + R_m(z_0, z) .$$

Putting $z = y - x$, expanding $(y - x - z_0)^\nu$ and reordering the terms results in

$$\sum_{|\nu|=0}^{m-1} \sum_{\mu=0}^{\nu} \kappa_{\nu,\mu-\nu}(z_0)\, y^\mu x^{\nu-\mu}$$

which is of the form (4.1).

Remark 4.1. For highly oscillatory kernel as, e.g., for the Helmholtz problem where $s(z) = e^{ik\|z\|}/\|z\|$, Taylor expansion is converging slowly. The situation can be improved by first performing a 1-d expansion in the radial direction, i.e.,

$$\frac{e^{ikr}}{r} \approx \frac{1}{r} \sum_{l=0}^{m-1} \frac{(ik)^l}{l!} e^{ikr_0} (r - r_0)^l$$

and then applying standard Taylor expansion to the right hand side above. The error term behaves like $\frac{((r-r_0)k)^m}{m!}$ and hence, a minimal requirement is that the order of the expansion has to satisfy

$$m \geq k(r - r_0) . \tag{4.2}$$

On the other hand the function system consists still of polynomials and the algorithmic apparatus developed for the Laplace operator can be used. An alternative to this expansion is the use of spherical Hankel functions, i.e., writing

$$\frac{e^{ikr}}{r} = \sum_{l=0}^{m-1} \alpha_l h_l(k(r - r_0))$$

which has a better convergence behaviour. However, (4.2) is the minimal requirement for convergence, too. For details see [15]. The use of this function system for Petrov-Galerkin discretizations makes the development of new algorithms necessary in order to compute the required quantities efficient.

In order to define the panel clustering algorithm, we have to define appropriate regions on the surface Γ where the expansion above gives good approximation.

Definition 4.1. *Let T_N denote the panelization of the Γ as, e.g., the triangulations explained in the previous chapters. A "cluster" is the union of one or more panels: $\tau = \bigcup_{j=1}^{q} \Delta_{i_j}$. The size of a cluster is given by the "cluster radius" $\rho(\tau)$ which is defined by the radius of the minimal ball containing τ. The centre z_τ of this balls is called "cluster center" .*

Definition 4.2. *The relative distance of a cluster from the support of a basis function of the test space $\varphi_i \in Y_n$ is given by*

$$d(\varphi_i, \tau) := \frac{\rho(\tau)}{\operatorname{dist}(\operatorname{supp}\varphi_i, z_\tau)},$$

where for the Dirac functional δ_x, i.e., the collocation method, $\operatorname{supp}\delta_x = x$.

Our aim is to determine the size of clusters such that the replacement of the kernel function by the expansion on those clusters has an accuracy of a given value $\epsilon > 0$. In this light, for $\epsilon > 0$ and arbitrary expansion order m, a cluster is said to be "admissible" with respect to a basis function $\varphi_i \in Y_n$, if

$$|k(x, y, y - x) - k_m(x, y)| \leq \epsilon \frac{1}{\|x - y\|}, \qquad \forall y \in \tau, x \in \operatorname{supp}\varphi_i. \tag{4.3}$$

The size of the clusters has to be linked to the approximation property of the kernel expansion in the following way.

Assumption 4.3. *For given ϵ and expansion order m, there exists $0 < \eta < 1$ such that, for all $1 \leq i \leq N$, the condition*

$$d(\varphi_i, \tau) \leq \eta < 1$$

implies that τ is admissible with respect to φ_i.

Definition 4.4. *Let ϵ and an expansion order $m \in \mathbb{N}$ be given, Let the relative size η of admissible clusters be determined as explained above. A set of clusters $\{\tau_i\}_{1 \leq i \leq k}$ with disjoint interiors is called a covering of Γ, if $\bigcup_{i=1}^{k} \tau_i = \Gamma$.*

A covering is called admissible with respect to a basis function $\varphi_i \in Y_n$, if either

$$d(\varphi_i, \tau) < \eta \qquad (\textit{admissible cluster})$$

or

$$\tau \textit{ is a panel.}$$

The admissible covering which contains a minimal number of clusters is called minimal admissible covering $\mathcal{C}_i$. The nearfield $\mathcal{C}_i^{near}$ and the farfield $\mathcal{C}_i^{far}$ are defined by

$$\mathcal{C}_i^{near} := \{\tau \in \mathcal{C}_i \mid \tau \textit{ is non-admissible with respect to } \varphi_i\},$$
$$\mathcal{C}_i^{far} := \{\tau \in \mathcal{C}_i \mid \tau \notin \mathcal{C}_i^{near}\}.$$

4.2 The Algorithm

Let $\boldsymbol{K}$ denote the matrix corresponding to Petrov-Galerkin discretization of the integral operator. Then, a matrix vector multiplication can be written in the form

$$\sum_{j=1}^{n} \boldsymbol{K}_{i,j}\,\boldsymbol{u}_j = \langle \varphi_i, K[u]\rangle = \left\langle \varphi_i,\ \text{p.f.} \int_\Gamma k(x,y,y-x)\,u(y)\,dy \right\rangle \tag{4.4}$$

with the dual pairing $\langle\cdot,\cdot\rangle$ of (2.3) and the correspondence (3.1) of a vector $\boldsymbol{u}$ and a FE-function u. Splitting the surface into the nearfield and the farfield part and using (4.1) shows that (4.4) equals

$$\begin{aligned}
&\sum_{j=1}^{n} \boldsymbol{u}_j \underbrace{\left\langle \varphi_i,\ \text{p.f.} \int_{\mathcal{C}_i^{near}} k(x,y,y-x)\,\phi_j(y)\,dy \right\rangle}_{=:\boldsymbol{K}_{i,j}^{near}} + \\
&\qquad = \sum_{\tau\in\mathcal{C}_i^{far}} \left\langle \varphi_i, \int_\tau k(x,y,y-x)\,u(y)\,dy \right\rangle \\
&\approx \sum_{j=1}^{n} \boldsymbol{K}_{i,j}^{near}\boldsymbol{u}_j + \sum_{\tau\in\mathcal{C}_i^{far}} \sum_{|\nu|=0}^{m-1} \sum_{\mu=0}^{\nu} \underbrace{\kappa_{\nu,\mu-\nu}(z_{i,\tau}) \left\langle \varphi_i, \left(\frac{\partial}{\partial n_x}\right)^{\alpha_1} \omega_\nu(x) \right\rangle}_{=:\kappa_{\nu,\mu-\nu}(\varphi_i,\tau)} \\
&\qquad \underbrace{\int_\tau \left(\frac{\partial}{\partial n_y}\right)^{\alpha_2} \omega_\mu(y)\,u(y)\,dy}_{J_\tau^\mu(u)} \\
&= \sum_{j=1}^{n} \boldsymbol{K}_{i,j}^{near}\boldsymbol{u}_j + \sum_{\tau\in\mathcal{C}_i^{far}} \sum_{|\nu|=0}^{m-1} \sum_{\mu=0}^{\nu} \kappa_{\nu,\mu-\nu}(\varphi_i,\tau)\,J_\tau^\nu(u)\,.
\end{aligned} \tag{4.5}$$

This is the panel clustering representation of a matrix vector multiplication. The coefficient κ are called the *expansion coefficient* and $J_\tau^\nu(u)$ the *farfield coefficient.*

The definition of the so-called nearfield matrix $\boldsymbol{K}_{i,j}^{near}$ looks identical as the original matrix $\boldsymbol{K}$. However, since the integral over $\mathcal{C}_i^{near}$ vanishes if $\text{supp}\phi_j \cap \mathcal{C}_i^{near} = \emptyset$, the matrix $\boldsymbol{K}_{i,j}^{near}$ is sparse dependent on the size of $\mathcal{C}_i^{near}$.

The panel clustering can be structured in the following way.

Phase I:

- Choose the accuracy ϵ for the kernel approximation.
- Choose the expansion order m and relative size of the admissible clusters $\eta = \eta(\epsilon, m)$ such that the storage/computational consumptions (which are known a priori) are minimal.

– Organize the cluster in a binary tree.
– For each basis function φ_i of Y_n, compute the near- and farfield of the minimal covering.

Phase II:

– Compute and store the nearfield matrix $\boldsymbol{K}_{i,j}^{\boldsymbol{near}}$, the expansion coefficients $\kappa_{\nu,\mu-\nu}(\varphi_i,\tau)$, and the *basis* nearfield coefficients $J_{\Delta}^{\nu}(\phi_i)$.

Phase III:

– Compute the farfield coefficients $J_{\tau}^{\nu}(u)$ from the coefficients $J_{\Delta}^{\nu}(\phi_i)$ by using the tree structure of the clusters.
– Approximate a matrix vector multiplication by (4.5).

5. Error Analysis of the Panel Clustering Method

The panel clustering method introduces a further error in the discretization process which can be treated similarly as, e.g., errors due to numerical cubature. In the present situation local analysis has to be done to determine the dependency of the accuracy of the kernel expansion on the size of the clusters and the order of the expansion. As a typical example, we consider Taylor approximation of the singularity function of the Laplace operator in 3-d. For given $\epsilon \geq 0$ and expansion order m, the bound for the relative size of admissible cluster η has to be chosen as $\eta = \sqrt[m]{\epsilon}$ in order to obtain

$$\left| \frac{1}{\|z\|} - T_m(z_0) \right| \leq \frac{\epsilon}{\|z\|}, \qquad \forall z : \frac{\|z - z_0\|}{\|z\|} \leq \eta$$

where T_m denotes the Taylor expansion of order m of $\|z\|^{-1}$. This estimate and also error estimates of Taylor-based expansions of general kernel functions are worked out in [17], [18], [8] and [13].

In the second step, the influence of these local errors (valid on the admissible clusters) to the consistency error, namely a matrix vector multiplication with and without panel clustering, has to be investigated. In [8], [18] and [9] it is shown that, under moderate assumption on the surface and the triangulation, the estimate

$$|\langle \varphi_i, (K - K_{PC})\, u \rangle| \leq C \epsilon \gamma_h \langle \varphi_i, 1 \rangle \|u\|_{\infty}, \qquad \forall u \in X_n,$$

is satisfied with γ_h equals 1 for weakly singular kernels, is $O(\log h)$ for Cauchy-singular and h^{-1} for hypersingular kernels. K_{PC} denote the panel clustering operator where the kernel is replaced by expansions satisfying Assumption 4.3 and (4.3).

5.1 Complexity of the Panel Clustering Algorithm

Let N denote the number of panels of the panelization T_N. An asymptotic complexity analysis of the panel clustering method (cf. [8]) shows that the choice of $m = \lfloor \frac{\kappa}{4} \log N \rfloor$ and $\eta = const$ implies that, again under moderate assumption on the surface and triangulation, the number of nearfield panels is bounded independent of N, while the number of farfield clusters grows like $O(\log N)$. Furthermore, the asymptotic amount of work for the initialization Phase II of the panel clustering algorithms is proportional to $O(N \log^r N)$ with $r = 5$ for the collocation and 4 for the Galerkin method. The evaluation Phase III requires $O(N \log^t N)$ with $t = 4$ for the collocation and $t = 7$ for the Galerkin method. The storage consumption for the collocation method are of order $O(N \log^4 N)$ and for the Galerkin BEM of order $O(N \log^3 N)$. The powers of the logarithmic terms are not ultimative. The algorithm can be structured such that Phase II becomes cheap and the storage consumptions are low (relatively few quantities are computed) but Phase III becomes expansive since more preparation steps have to be performed and vice versa.

The asymptotic gain of the panel clustering method is obvious compared to the standard matrix techniques where both the CPU-time and the storage consumption behaves like $O(N^2)$. Nevertheless it is important to consider the constants of the $O(\cdot)$ estimates in detail to study the amount of work used by the panel clustering algorithm for "small" problem sizes. As a test problem we examined the Galerkin discretization of the double-layer potential for the Laplace equation on the surface of the unit cube scaled by $(3 : 1 : 0.3)$. For the panel clustering and the conventional matrix oriented version the amount of time and space required to assemble the corresponding discrete operator $\tilde{\boldsymbol{K}}$ satisfying a given relative error ϵ, i.e., $\|\boldsymbol{K}\boldsymbol{u} - \tilde{\boldsymbol{K}}\boldsymbol{u}\|_2 \leq \epsilon \|\boldsymbol{K}\boldsymbol{u}\|_2$ with $\boldsymbol{K}$ the exact matrix, was measured. The accuracy of $\tilde{\boldsymbol{K}} = \tilde{\boldsymbol{K}}^{\mathbf{pc}}$ of the panel clustering was controlled by the parameter η whereas m was fixed. In order to adjust the accuracy of $\tilde{\boldsymbol{K}} = \tilde{\boldsymbol{K}}^{\mathbf{m}}$ of the matrix oriented version the size of the nearfield, where high order cubature formulae are used, and the farfield, where only one-point formulae are applied, was changed. The results for a problem size of $N = 1154$ vertices (2304 panels) are shown in Figure 5.1 illustrating the amount of time needed to assemble the operators and in Figure 5.2 depicting the number of entries to represent these operators. The calculations were carried out on a SUN SPARC-10/40. It turns out that even for small problem sizes the panel clustering method could reduce the CPU-time and storage consumptions considerably. Further results are reported in [18] and [13].

6. Software Design Aspects

Regarding the software activities for finite element methods (FEM) there are already complex and powerful packages to solve various problems. It

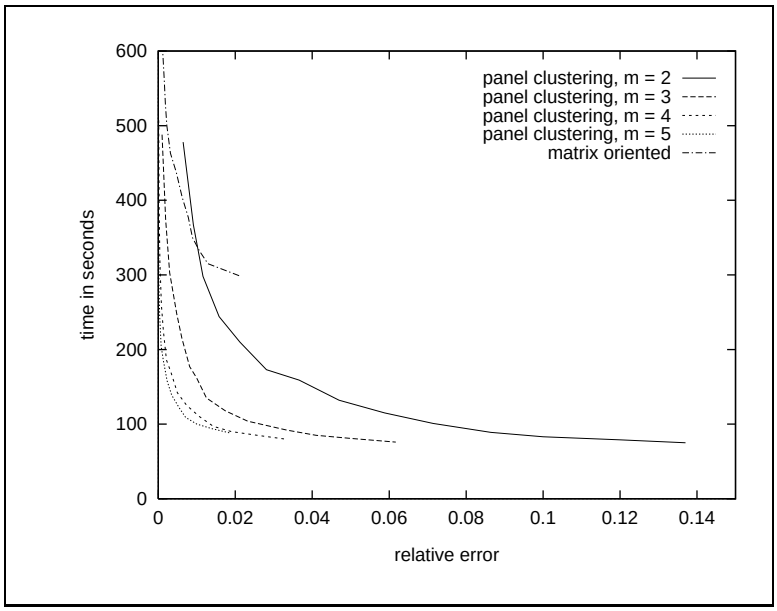

Fig. 5.1. Time consumptions assembling the discrete operators.

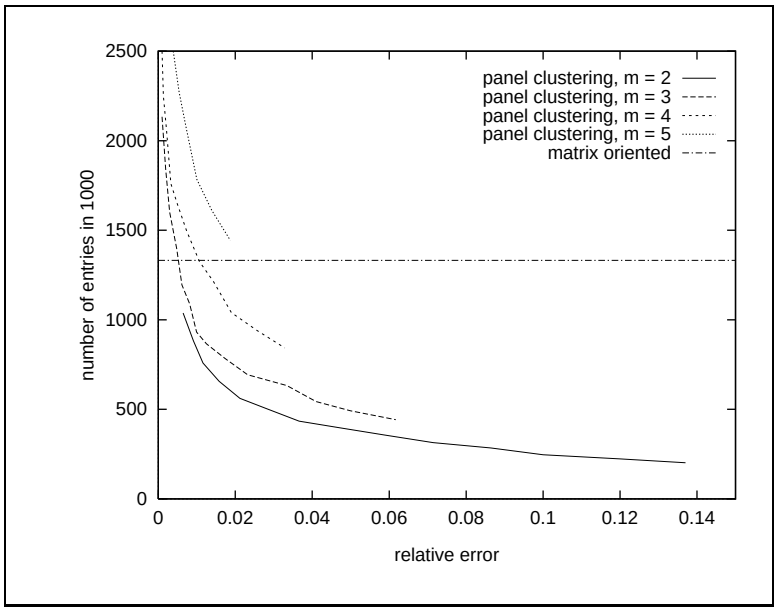

Fig. 5.2. Storage consumptions of the discrete operators.

would be quite convenient to reuse this software or at least the design aspects for the boundary element method as well. Though even the conception of FEM and BEM are nearly the same, the aspects concerning design decisions differ considerably. The system matrices of the FEM, for instance, are sparse. Adequate data structures for this type of matrices are lists or other containers taking the local character into consideration. However, the BEM leads to full matrices making it quite plain to use arrays. There are a lot of other examples like this one showing that it is unavoidable to develop BEM software from the beginning.

The difficulty one has to encounter is the great variety of tasks: There are among others different discretization schemes like collocation or Galerkin methods, different kinds of operators (single layer, double layer, hypersingular) and quite a lot of cubature techniques to consider. A design strategy which isolates the common parts and 'focus upon the essential characteristics' [3] to build the basic abstract data types would be very useful. These properties are given by object-oriented methods known to be a suitable tool, to manage complex systems. In the following we specify an object-oriented approach to BEM which has been elaborated as a class library implemented in C++.

In our situation the object-oriented decomposition could be extracted from the formulation of Petrov-Galerkin schemes introduced in Chapter 2.. The foundation of this scheme are the subspaces X_n and Y_n defining the test and trial functions. A basic information necessary to construct these spaces is a description of the geometry, i.e. the surfaces. For this purpose it is advantageous to have a (continuous) description which offers the possibility to resolve the geometry with any necessary accuracy – an important fact, especially in the case of multilevel methods where hierarchies of spaces must be generated. So we can state a first class of the library called `Surface` representing a formal description of the surface:

```
class Surface {
        :
    public:
      Surface(char* fname);
        :
};
```

```
class SurfaceApx {
        :
    public:
      SurfaceApx(const Surface& sf, int level);
        :
};
```

In the next step, we require an approximation of the surface, i.e. a set of panels (triangulation) that meets several conditions listed in Chapter 2.. To maintain this set, a second class, called `SurfaceApx`, has to be introduced into the context of geometry abstraction. In the class definition above the parameter `level` is used to distinguish between particular levels of approximation. Instances of the class `SurfaceApx` for different levels form a hierarchy of panels.

To classify a basis of the subspaces two well known alternatives are available. The first one, the node-oriented version, characterizes a single basis function by its support in conjunction with a related node. The basis is described by the set of all basis functions. This definition is obvious and close to the discrete mathematical formulation of the Petrov-Galerkin scheme. Nevertheless the second, panel-oriented alternative is more efficient in respect of the time-consuming integrations. It uses the corresponding shape function for each panel, i.e., the set of all non-vanishing basis functions restricted to the panel. The abstraction of the entire basis is formed by the recombination of all shape functions. An entity, representing a single basis function, no longer exists.

The properties of common shape functions are reflected in the definition of an abstract base class:

```
class ShapeFunction {
        :
      public:
        virtual const Panel&   support() const = 0;
        virtual int    dimension() const = 0;
        virtual const int*   index() const = 0;
          :
};
```

It gives information about the related panel, the number of restricted basis functions and their associated indices. By the mechanism of inheritance we can generate subclasses of `ShapeFunction` to module different basis functions, as constant, linear or dirac functions, on different types of panels (Figure 6.1).

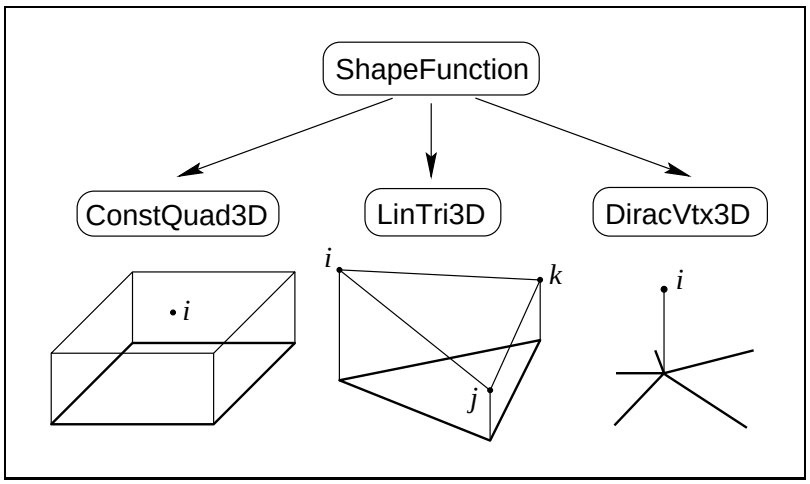

Fig. 6.1. Specializations of `ShapeFunction`.

For the instantiation and maintenance of shape functions a further class, `Space`, is established. This provider is responsible for the correct combination of shape functions to form a basis and thus the desired subspace of the Petrov-Galerkin method; especially it has to support the management of conforming indices. To realize various subspaces, several specializations of the base class `Space` are used. Analogous to the subclasses of `Shapefunction` we could define `ConstSpace`, `LinSpace` or `DiracSpace`.

The next class in our library addresses the dual forms of (2.3). Corresponding to the construction of spaces the abstraction of a dual form is a function that assigns to each pair of shape functions (Φ, Ψ) an element matrix of the form

$$(m)_{\phi\psi} := \langle K\psi, \phi \rangle , \quad \phi \in \Phi, \psi \in \Psi.$$

Embeded in a class definition we get

```
class DualForm {
  public:
    virtual const real* operator()(const ShapeFunction& trialfnc,
                                   const ShapeFunction& testfnc) const;
};
```

Before the assignment can be executed an appropriate integration or cubature rule dependent on the particular type of shape functions has to be chosen, e.g., to distinguish between collocation and Galerkin schemes. For this purpose run-time type information (RTTI) is an essential tool and used in our implementation to solve this kind of multi-method problem (see [23]). The class `DualForm` acts as an interface class which 'adjusts the appearance' [23] for further class definitions derived from it. With these subclasses a variety of integral operators and associated integrations strategies could be arranged.

To assemble local element matrices in a global one the class `MatrixOp` is introduced into the design. It is a specialization of the base class `Operator` (Figure 6.2) which first of all is responsible for the abstraction of integral operators. This is done by defining a virtual member function of the class `Operator` common to all subclasses which declares the mapping of the considered operator:

```
virtual void operator()(const Function& trialfnc, const Function& testfnc);
```

For the class `MatrixOp` for instance this function is to be overloaded by a simple matrix-vector product. The class `Function` in the definition above just represents a valuation of the related basis. It can be implemented by a vector of floats or doubles.

The subclass `MatrixOp` is used to specify the integral operator on the discrete level by a full coefficient matrix. Another alternative offers the panel clustering method described in Chapter 4.. Its complex algorithmic issues can be totaly hidden in the implementation of an additional subclass of `Operator` denoted by `PnlClstOp`. The external behaviour or usage resp. of `PnlClstOp` is nearly the same as for the simple `MatrixOp`. Only the information needed during the instantiation process must be extended: a specialized version of

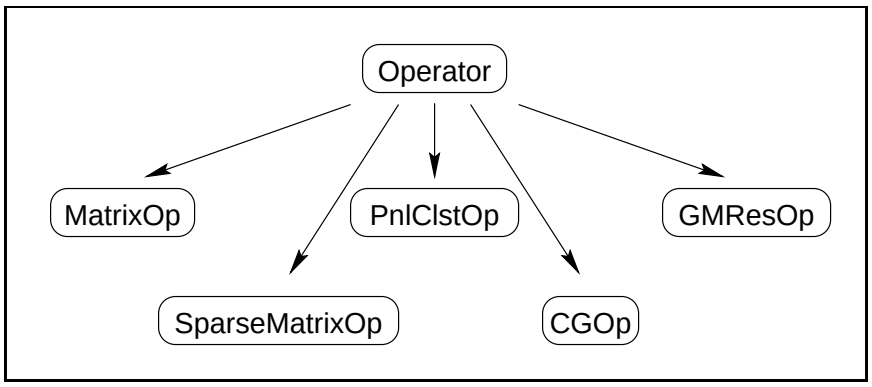

Fig. 6.2. Specializations of `Operator`.

the class **`DualForm`** to describe the far field dual forms $\langle \partial^{\alpha_1}/\partial n_x \, w_\mu(x), \varphi_i \rangle$ and $\langle \phi_i, \partial^{\alpha_2}/\partial n_y \, w_\nu(y) \rangle$ as well as an instruction to calculate the expansion coefficients $\kappa_\nu(z)$.

Even the solver of the linear system interpreted as inverse operator meets this context. For example overloading the `operator()` by the gmres algorithm leads to the class **`GMResOp`**. An instance of **`GMResOp`** is initialized with the operator that should be inverted.

To summarize the rough introduction of our class library we present a short application:

```
int main() {
  // initialize the geometry given in file "cube"
  Surface       sf("cube");
  SurfaceApx    sfapx(sf, 2);

  // construct test and trial spaces for collocation
  DiracSpace    test(sfapx);
  LinSpace      trial(sfapx);

  // get the necessary informations about dual forms and
  // expansion coefficients
  DlpDF             df(3);
  FarFieldDF        ftstdf(2);
  FarFieldDrvdDF    ftrldf(2);
  LaplaceExp        exp(2);

  // assemble the stiffness matrix and the panel clustering operator
  MatrixOp    op(trial, test, df);
  PnlClstOp   pcop(trial, test, df, ftrldf, ftstdf, exp, 1-e2);

  // initialize two functions; one matching the trial space,
  // another the test space
  Function f(trial); f = "sin(x*y)";
  Function g(test);

  // calculate (⟨Kf, φ_i⟩)_j with both operators
```

```
    op(f, g);
    pcop(f, g);

    // solve using the panel cluster operator
    GMResOp solve(pcop, 1e-8, 100);
    solve(g, f);
}
```

Finally the approach to involve parallel architectures, in particular SPMD schemes (same program multiple data), should be mentioned. The most time-consuming task of the BEM is to assemble the matrices. A distribution of this task among several processors can be achieved by decomposing the subspaces X_n and Y_n leading to a two dimensional torus as proper topology of processors (Figure 6.3). To be independent of the given physical architecture an interface class `ProcessNode` is created which establishs basic operations in the terms of the underlying torus (e.g. to broadcast along rows or columns, the upper, lower, left or right neighbour):

```
class ProcessNode {
        :
  public:
    virtual int broadcast(...) = 0;
    virtual int reduce(...) = 0;
    virtual int shift(...) = 0;
        :
};
```

Specialisations of this interface class realize the connection between the physical layer (hypercube, workstation cluster, etc.) and the application layer (2d torus).

To make the information and operations of `ProcessNode` available to the concerned classes of the library a reference to an instance of `ProcessNode` could be used. The virtual function call ensures the expected behaviour. Parameterized classes (templates) offer another, more efficient way of embedding the derived classes of `ProcessNode` in the library. In this alternative the member functions `broadcast()`, `reduce()`, etc. are bounded statically resulting in an efficient function call.

References

1. K. Atkinson. Solving Integral Equations on Surfaces in Space. In G. Hämmerlin and K. Hoffmann, editors, *Constructive Methods for the Practical Treatment of Integral Equations*, pages 20–43. Birkhäuser: ISNM, 1985.
2. K. Atkinson and D. Chien. Piecewise Polynomial Collocation for Boundary Integral Equations. *J. Sci. Comp.*, 16:651-681, 1995.
3. G. Booch *Object-oriented Analysis and Design*, Second Edition Benjamin/-Cummings Publishing Company, Redwood City, California, 1994.

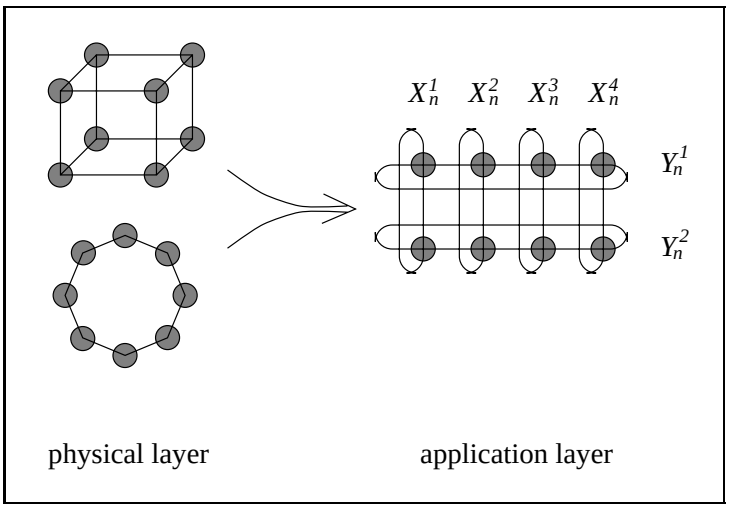

Fig. 6.3. Abstraction of parallel architectures.

4. A. Brandt. Multilevel computations of integral transforms and particle interactions with oscillatory kernels. In *IMACS 1st Int. Conf. On Comp. Phys., Boulder, Colorado*, June 1990.
5. A. Brandt and A. Lubrecht. Multilevel Matrix Multiplication and Fast Solutions of Integral Equations. *J. Comp. Physics*, 90:348–370, 1990.
6. D. Dahmen, S. Prößdorf, and R. Schneider. Wavelet Approximation Methods for Pseudodifferential Equations II: Matrix Compression and Fast Solutions. *Advances in Comp. Math.*, 1:259–335, 1993.
7. W. Hackbusch. The Solution of Large Systems of BEM Equations by the Multi-Grid and Panel Clustering Technique. *Rend. Sem. Mat. Uni. Pol. Torino, Fasc. Spec.: Numerical Methods*, pages 163–187, 1991.
8. W. Hackbusch and Z. Nowak. On the Fast Matrix Multiplication in the Boundary Element Method by Panel-Clustering. *Numerische Mathematik*, 54:463–491, 1989.
9. W. Hackbusch and S. Sauter. On the Efficient Use of the Galerkin Method to Solve Fredholm Integral Equations. *Application of Mathematics (formerly: aplikace matematiky)*, 38(4-5):301–322, 1993.
10. W. Hackbusch and S. A. Sauter. On Numerical Cubatures of Nearly Singular Surface Integrals arising in BEM Collocation. *Computing*, 52:139–159, 1994.
11. F. John. *Plane Waves and Spherical Means*. Springer Verlag, New York, 1955.
12. R. Kieser. *Über einseitige Sprungrelationen und hypersinguläre Operatoren in der Methode der Randelemente*. PhD thesis, Mathematisches Institut A, Universität Stuttgart, Germany, 1990.
13. C. Lage. *Analyse, Entwurf und Implementation von Randelementmethoden*. PhD thesis, Inst. f. Prakt. Math., Universität Kiel, to appear in 1995.
14. V. Rokhlin. Rapid solutions of integral equations of classical potential theory. *Journal of computational Physics*, 60(2):pp. 187–207, 1985.
15. V. Rokhlin. Diagonal Forms of Translation Operators for the Helmholtz Equation i Three Dimensions. *Appl. and Comp. Harm. Anal.*, 1(1):82-93, 1993

16. S. Sauter and A. Krapp. On the Effect of Numerical Integration in the Galerkin Boundary Element Method. Technical Report 95-4, Lehrstuhl Praktische Mathematik, Universität Kiel, Germany, 1995.
17. S. A. Sauter. Der Aufwand der Panel-Clustering-Methode für Integralgleichungen. Technical Report 9115, Institut für Praktische Mathematik, University of Kiel, 24118 Kiel, Germany, 1991.
18. S. A. Sauter. *Über die effiziente Verwendung des Galerkinverfahrens zur Lösung Fredholmscher Integralgleichungen*. PhD thesis, Inst. f. Prakt. Math., Universität Kiel, 1992.
19. R. Schneider, T. v.Petersdorff, and C. Schwab. Multiwavelets for Second Kind Integral Equations. *SIAM, J. Numer. Anal.*, to appear.
20. C. Schwab. Variable Order Composite Quadrature of Singular and Nearly Singular Integrals. *Computing*, 53:173-194, 1995.
21. C. Schwab and W. Wendland. Kernel Properties and Representations of Boundary Integral Operators. *Math. Nachr.*, 156:187–218, 1992.
22. C. Schwab and W. Wendland. On Numerical Cubatures of Singular Surface Integrals in Boundary Element Methods. *Numerische Mathematik*, pages 343–369, 1992.
23. B. Stroustrup. *The C++ Programming Language*, Second Edition Addison-Wesley Publishing Company, Reading Massachusetts, 1991.
24. B. Stroustrup. *The Design and Evolution of C++* Addison-Wesley Publishing Company, Reading Massachusetts, 1994.