Acta Numerica (2006), pp. 543–639
doi: 10.1017/S0962492906270016

Kernel techniques: From machine learning to meshless methods

Robert Schaback and Holger Wendland
Institut für Numerische und Angewandte Mathematik,
Universität Göttingen, Lotzestraße 16–18,
D–37083 Göttingen, Germany
E-mail: {schaback}{wendland}@math.uni-goettingen.de
http://www.num.math.uni-goettingen.de/schaback
http://www.num.math.uni-goettingen.de/wendland

Kernels are valuable tools in various fields of numerical analysis, including approximation, interpolation, meshless methods for solving partial differential equations, neural networks, and machine learning. This contribution explains why and how kernels are applied in these disciplines. It uncovers the links between them, in so far as they are related to kernel techniques. It addresses non-expert readers and focuses on practical guidelines for using kernels in applications.

CONTENTS

1. Introduction

This article can be seen as an extension of Martin Buhmann's presentation of *radial basis functions* (Buhmann 2000) in this series. But we shall take a somewhat wider view and deal with *kernels* in general, focusing on their recent applications in areas such as *machine learning* and *meshless methods* for solving partial differential equations.

In their simplest form, kernels may be viewed as bell-shaped functions like Gaussians. They can be shifted around, dilated, and superimposed with weights in order to form very flexible spaces of multivariate functions having useful properties. The literature presents them under various names in contexts of different numerical techniques, for instance as *radial basis functions*, *generalized finite elements*, *shape functions* or even *particles*. They are useful both as *test functions* and *trial functions* in certain *meshless methods* for solving partial differential equations, and they arise naturally as *covariance kernels* in probabilistic models. In the case of learning methods, sigmoidal functions within neural networks were successfully superseded by radial basis functions, but now they have both been replaced by *kernel machines*[1] to implement the most successful algorithms for *machine learning* (Schölkopf and Smola 2002, Shawe-Taylor and Cristianini 2004). Even the term *kernel engineering* has been coined recently, because efficient learning algorithms require specially tailored application-dependent kernels.

With this slightly chaotic background in mind, we survey the major application areas while focusing on a few central issues that lead to guidelines for practical work with kernels. Section 2 starts with a general definition of kernels and provides a short account of their properties. The main reasons for using kernels at all will be described in Section 3, starting with their ability to recover functions optimally from given unstructured data. At this point, the connections between kernel methods for interpolation, approximation, learning, pattern recognition, and PDE solving become apparent. The probabilistic aspects of kernel techniques follow in Section 4, while practical guidelines for constructing new kernels follow in Section 5. Special application-oriented kernels are postponed to Section 6 to avoid too much detail at the beginning.

Since one of the major features of kernels is to generate spaces of trial functions with excellent approximation properties, we devote Section 7 to a short account of the current results concerning such questions. Together with strategies to handle large and ill-conditioned systems (Section 8), these results are of importance to the applications that follow later.

After a short interlude on kernels on spheres in Section 9 we start our survey of applications in Section 10 by looking at interpolation problems

[1] `http://www.kernel-machines.org`

first. These take advantage of the abilities of kernels to handle unstructured Birkhoff-type data while producing solutions of arbitrary smoothness and high accuracy. Then we review kernels in modern learning algorithms, but we can keep this section short because there are good books on the subject.

In Section 12 we survey meshless methods (Belytschko, Krongauz, Organ, Fleming and Krysl 1996*b*) for solving partial differential equations. We describe the different techniques currently sailing under this flag, and point out where and how kernels occur. Owing to an existing survey (Babuška, Banerjee and Osborn 2003) in this series, we keep the generalized finite element method short here, but we incorporate meshless local Petrov–Galerkin techniques (Atluri and Shen 2002).

The final two sections then focus on purely kernel-based meshless methods. We treat applications of symmetric and unsymmetric collocation, of kernels providing fundamental and particular solutions, and provide the state of the art of their mathematical foundation.

Altogether, we want to keep this survey digestible for the non-expert and casual reader who wants to know roughly what has happened so far in the area of application-oriented kernel techniques. This is why we omit most of the technical details and focus on the basic principles. Consequently, we have to refer as much as possible to background reading for proofs and extensions. Fortunately, there are two recent books, Buhmann (2004) and Wendland (2005*b*), which contain the core of the underlying general mathematics for kernels and radial basis functions. For kernels in learning theory, we have already cited two other books, Schölkopf and Smola (2002) and Shawe-Taylor and Cristianini (2004), providing further reading. If we omit pointers to proofs, these books will contain what is needed.

Current books and survey articles in the area of meshless methods are rather specialized, because they focus either on certain classes of methods or on applications. We cite them as needed, placing them into a more general context. Clearly, the list of references cannot contain all available papers on all possible kernel applications. This forces us to select a very small subset, and our main selection criterion is how a certain reference fits into the current line of argument at a certain place of this survey. Incorporation or omission of a certain publication does not express our opinion on its importance in general.

2. Kernels

Definition 2.1. A *kernel* is a function

$$K\;:\;\Omega\times\Omega\to\mathbb{R}$$

where Ω can be an arbitrary nonempty set.

Some readers may consider this to be far too general. However, in the context of learning algorithms, the set Ω defines the possible *learning inputs.* Thus Ω should be sufficiently general to allow Shakespeare texts or X-ray images, *i.e.*, Ω should preferably have no predefined structure at all. Thus the kernels occurring in machine learning are extremely general, but they still take a special form which can be tailored to meet the demands of applications. We shall now explain the recipes for their definition and usage.

2.1. Feature maps

In certain situations, a kernel is given *a priori*, *e.g.*, the *Gaussian*

$$K(x,y) := \exp(-\|x-y\|_2^2) \quad \text{for all } x,y \in \Omega := \mathbb{R}^d. \tag{2.1}$$

Each specific choice of a kernel has a number of important and possibly unexpected consequences which we shall describe later.

If no predefined kernel is available for a certain set Ω, an application-dependent *feature map* $\Phi : \Omega \to \mathcal{F}$ with values in a Hilbert '*feature*' space $\mathcal{F}$ is defined. It should provide for each $x \in \Omega$ a large collection $\Phi(x)$ of *features* of x which are characteristic for x and which live in the Hilbert space $\mathcal{F}$ of high or even infinite dimension. Note that $\mathcal{F}$ has plenty of useful structure, while Ω does not.

Guideline 2.2. Feature maps $\Omega \to \mathcal{F}$ allow us to apply linear techniques in their range $\mathcal{F}$, while their domain Ω is an unstructured set. They should be chosen carefully in an application-dependent way, capturing the essentials of elements of Ω.

With a feature map Φ at hand, there is a *kernel*

$$K(x,y) := \big(\Phi(x), \Phi(y)\big)_{\mathcal{F}} \quad \text{for all } x,y \in \Omega. \tag{2.2}$$

In another important class of cases, the set Ω consists of random variables. Then the *covariance* between two random variables x and y from Ω is a standard choice of a kernel. These and other kernels arising in nondeterministic settings will be the topic of Section 4. The connection to learning is obvious: two learning inputs x and y from Ω should be very similar, if they are closely 'correlated', if they have very similar features, or if (2.2) takes large positive values. These examples suggest the following definition.

Definition 2.3. A kernel K is *symmetric* if $K(x,y) = K(y,x)$ holds for all $x, y \in \Omega$.

2.2. Spaces of trial functions

A kernel K on Ω defines a function $K(\cdot, y)$ for all fixed $y \in \Omega$. This allows us to generate and manipulate spaces

$$\mathcal{K}_0 := \operatorname{span}\{K(\cdot,y) \ : \ y \in \Omega\} \tag{2.3}$$

of functions on Ω. In learning theory, the function $K(\cdot, y) = (\Phi(\cdot), \Phi(y))_{\mathcal{F}}$ relates each other input object to a fixed object y via its essential features. But in general $\mathcal{K}_0$ just provides a handy linear space of *trial* functions on Ω which is extremely useful for most applications of kernels, *e.g.*, when Ω consists of texts or images. For example, in meshless methods for solving partial differential equations, certain finite-dimensional subspaces of $\mathcal{K}_0$ are used as *trial* spaces to furnish good approximations to the solutions.

2.3. Convolution kernels

In certain other cases, the set Ω carries a measure μ, and then, under reasonable assumptions like $f,\ K(y,\cdot) \in L^2(\Omega, \mu)$, the generalized *convolution*

$$K *_\Omega f := \int_\Omega f(x) K(\cdot, x)\,\mathrm{d}\mu(x) \tag{2.4}$$

defines an integral transform $f \mapsto K *_\Omega f$ which can be very useful. Note that Fourier or Hankel transforms arise this way, and recall the role of the Dirichlet kernel in the Fourier analysis of univariate periodic functions. The above approach to kernels via convolution works on locally compact topological groups using Haar measure, but we do not want to pursue this detour into abstract harmonic analysis too far. For space reasons, we also have to exclude complex-valued kernels and all transform-type applications of kernels here, but it should be pointed out that wavelets are special kernels of the above form, defining the *continuous wavelet transform* this way.

Note that discretization of the integral in the convolution transform leads to functions in the space $\mathcal{K}_0$ from (2.3). Using kernels as trial functions can be viewed as a discretized convolution. This is a very useful fact in the theoretical analysis of kernel-based techniques.

Guideline 2.4. Kernels have three major application fields: they generate convolutions, trial spaces, and covariances. The first two are related by discretization.

2.4. Scaling

Another important aspect in all kernel-based techniques is the *scaling problem.* If the kernel K in the convolution equation (2.4) is a sharp nonnegative spike with integral one, the convolution will reproduce f approximately, and the distributional 'delta kernel' will reproduce f exactly. This is theoretically nice, but discretization will need a very fine spatial resolution. On the other hand, convolution with a nonnegative smooth kernel of wide or infinite support acts as a *smoothing operator* which will not have good reproduction quality. To control this trade-off between approximation and smoothing, many kernel applications involve a free scaling parameter, and it is a serious problem to derive good strategies for its determination.

Guideline 2.5. Success and failure of kernel usage may crucially depend on proper scaling.

The scaling problem will come up at various places in this article.

2.5. Positive definiteness

For many applications, the space $\mathcal{K}_0$ needs more structure. In fact, it can be turned into a Hilbert space via the following construction.

Definition 2.6. A symmetric kernel K is *positive (semi-) definite* if, for all finite subsets $X := \{x_1, \ldots, x_N\}$ of distinct points of Ω, the symmetric *kernel matrices* $A_{K,X}$ with entries $K(x_j, x_k)$, $1 \le j,k \le N$ are positive (semi-) definite.

We delay the definition of *conditionally* positive definite kernels to Section 6. For a symmetric positive definite kernel K on Ω, the definition

$$(K(x,\cdot), K(y,\cdot))_{\mathcal{K}} = (K(\cdot,x), K(\cdot,y))_{\mathcal{K}} := K(x,y) \quad \text{for all } x,y \in \Omega \tag{2.5}$$

of an inner product of two generators of $\mathcal{K}_0$ easily generalizes to an inner product on all of $\mathcal{K}_0$ such that

$$\left\| \sum_{j=1}^N \alpha_j K(\cdot, x_j) \right\|_{\mathcal{K}}^2 := \sum_{j,k=1}^N \alpha_j \alpha_k K(x_j, x_k) = \alpha^T A_{K,X} \alpha \tag{2.6}$$

defines a *numerically accessible* norm on $\mathcal{K}_0$ which allows us to construct a *native* Hilbert space

$$\mathcal{K} := \operatorname{clos} \mathcal{K}_0 \tag{2.7}$$

as the completion, or Hilbert space closure, of $\mathcal{K}_0$ under the above norm. In most cases, the space $\mathcal{K}$ is much richer than $\mathcal{K}_0$ and does not seem to have any explicit connection to the kernel from which it is generated. For instance, Sobolev spaces $\mathcal{K} = W_2^k(\mathbb{R}^d)$ with $k > d/2$ result from the kernel

$$K(x,y) = \|x-y\|_2^{k-d/2} K_{k-d/2}(\|x-y\|_2) \tag{2.8}$$

where K_ν is the Bessel function of third kind. Starting from (2.8) it is not at all clear that the closure (2.7) of the span (2.3) of all translates of K generates the Sobolev space $W_2^k(\mathbb{R}^d)$. But it should be clear that the native Hilbert space for a kernel has important consequences for any kind of numerical work with the trial space $\mathcal{K}_0$ of (2.3).

Guideline 2.7. User of kernel techniques should always be aware of the specific native Hilbert space associated to the kernel.

Under certain additional assumptions, there is a one-to-one correspondence between symmetric positive definite kernels and Hilbert spaces of functions, so that such kernels cannot be separated from their native Hilbert space.

However, note that in general the Hilbert spaces $\mathcal{F}$ from (2.2) and $\mathcal{K}$ from (2.5) are different. The space $\mathcal{K}$ is always a Hilbert space of functions on Ω, while the 'feature space' $\mathcal{F}$ in general is not. However, the two notions coincide if we start with a given kernel, not with a feature map.

Theorem 2.8. Every symmetric positive definite kernel can be generated via a suitable feature map.

Proof. Given a symmetric positive definite kernel K, define $\Phi(x) := K(x,\cdot)$ and $\mathcal{F} := \mathcal{K}$ using (2.5) to get (2.2). □

2.6. Reproduction

By construction, the spaces $\mathcal{K}$ and $\mathcal{K}_0$ have a nice structure now, and there is a *reproduction property*

$$f(x) := (f, K(\cdot, x))_{\mathcal{K}} \quad \text{for all } f \in \mathcal{K},\ x \in \Omega \tag{2.9}$$

for all functions in $\mathcal{K}$. At this point, we are on the classical ground of *reproducing kernel Hilbert spaces* (RKHS) with a long history (Aronszajn 1950, Meschkowski 1962, Atteia 1992).

Guideline 2.9. Positive definite kernels *reproduce* all functions from their associated *native* Hilbert space. On the trial space (2.3) of translated positive definite kernels, the Hilbert space norm can be *numerically calculated* by plain kernel evaluations, without integration or derivatives. This is particularly useful if the Hilbert space norm theoretically involves integration and derivatives, *e.g.*, in the case of Sobolev spaces.

2.7. Invariance

Guideline 2.10. If the set Ω has some additional geometric structure, kernels may take a simplified form, making them *invariant* under geometric transformations on Ω.

For instance, kernels of the form

$K(x-y)$	are *translation-invariant* on abelian groups
$K(x^Ty)$	are *zonal* on multivariate spheres
$K(\|x-y\|_2)$	are *radial* on $\mathbb{R}^d$

with a slight abuse of notation. Radial kernels are also called *radial basis functions*, and they are widely used because of their invariance under Euclidean (rigid-body-) transformations in $\mathbb{R}^d$. The most important example is the *Gaussian* kernel of (2.1), which is symmetric positive definite on $\mathbb{R}^d$, for any space dimension d. It naturally arises as a convolution kernel, a covariance, a perfectly smooth trial function, and a multivariate probability density, illustrating the various uses of kernels. Less obvious is the fact that it has a native Hilbert space of analytic functions on $\mathbb{R}^d$.

2.8. Metric structure

In many applications, for instance in machine learning, the kernel value $K(x,y)$ increases with the 'similarity' of x and y, like a correlation or a covariance, and is *bell-shaped* like the Gaussian. More precisely, any symmetric positive definite kernel K generates a *distance metric* $d : \Omega\times\Omega \to [0,\infty)$ via

$$d^2(x,y) := K(x,x) - 2K(x,y) + K(y,y) \quad \text{for all } x,y \in \Omega \tag{2.10}$$

on a general set (Schoenberg 1937, Stewart 1976). Looking back at feature maps, we see that a well-chosen feature map defines a kernel that introduces a metric structure on the set Ω for which 'close' elements have 'similar' features.

Guideline 2.11. Symmetric positive definite kernels on Ω introduce a 'geometry' on the set Ω which can be tailored to meet the demands of applications.

The art of *kernel engineering* is to do this in a best possible way, depending on the application in question.

3. Optimal recovery

One of the key advantages of kernels is as follows.

Guideline 3.1. Kernel-based methods can make optimal use of the given information.

Results like this come up at various places in theory and applications, and they have a common background linking them to the interesting fields of *information-based complexity*[2] (Traub and Werschulz 1998) and *optimal recovery* (Micchelli, Rivlin and Winograd 1976, Micchelli and Rivlin 1977) which we have to ignore here. In a probabilistic context, Guideline 3.1 can be forged into an exact statement using Bayesian arguments, but we want to keep things simple first and postpone details to Section 4.

3.1. Recovery from unstructured data

Assume that we want to model a black-box transfer mechanism like Figure 3.1 that replies to an input $x \in \Omega$ by an output $f(x) \in \mathbb{R}$. This can be the reaction $f(x)$ of a well-trained individual or machine to a given stimulus

[2] `http://www.ibc-research.org`

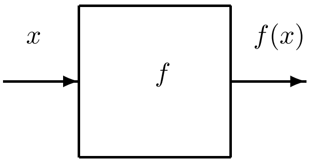

Figure 3.1. A black-box response mechanism.

x given to it. Finding a good response mechanism f can be called *learning* or *black-box modelling.* If the output should take only a finite number of possible values, this is *pattern recognition* or *classification.* We shall use the term 'recovery problem' (Micchelli *et al.* 1976, Micchelli and Rivlin 1977) to summarize all of these situations, which mathematically require the determination of a function. But we want to stick to an application-oriented view here.

At this point we do not have any further information on the model or the intended reactions to the stimuli. But usually we have some examples of 'good behaviour' that can be used. These take the form of a sequence $(x_1, y_1), \dots, (x_N, y_N)$ of unstructured *training data*, pairing inputs $x_j \in \Omega$ with their expected responses $y_j \in \mathbb{R}$. The recovery task now is to find a function f such that

$$f(x_j) \approx y_j, \quad 1 \le j \le N, \tag{3.1}$$

and this is a standard interpolation or approximation problem, though posed on an unstructured set Ω using unstructured data.

If we slightly extend the meaning of the word 'data', we can try to find a smooth function f such that

$$\begin{aligned} (-\Delta f)(y_j) &\approx \varphi(y_j), \quad 1 \le j \le M, \\ f(z_k) &\approx \psi(z_k), \quad M+1 \le k \le N, \end{aligned} \tag{3.2}$$

where $y_1, \dots, y_M$ are points in a bounded domain Ω while $z_{M+1}, \dots, z_N$ lie on the boundary. This would hopefully provide an approximate solution f to the Poisson problem

$$\begin{aligned} (-\Delta f)(y) &= \varphi(y), \quad y \in \Omega, \\ f(z) &= \psi(z), \quad z \in \partial\Omega, \end{aligned}$$

for given functions φ on Ω and ψ on $\partial\Omega$. Note that this *collocation* technique is again a recovery problem for a function f from certain of its data, just replacing point evaluations in (3.1) by evaluations of certain derivatives. In general, one can replace (3.1) by

$$\lambda_j(f) \approx y_j, \quad 1 \le j \le N, \tag{3.3}$$

for a set of given linear *data functionals* $\lambda_1, \dots, \lambda_N$ generalizing the point evaluation functionals $\delta_{x_1}, \ldots, \delta_{x_N}$ of (3.1). Tradition in approximation theory would call this a recovery problem from *Hermite–Birkhoff* data, if the data functionals are evaluations of derivatives at points. But there are much more general functionals, *e.g.*, those defining *weak data* via

$$\lambda_j(f) = \int_\Omega \nabla f \cdot \nabla v_j$$

as in finite elements, using a *test function* v_j. This way, finite element methods for solving linear partial differential equations can be written as recovery problems (3.3).

For later sections of this article, the reader should keep in mind that suitable generalizations (3.3) of the recovery problem (3.1) lead to methods for solving partial differential equations. We shall stick to the simple form of (3.1) for a while, but when reviewing large parts of numerical analysis, *e.g.*, finite element techniques, we have the following heuristic.

Guideline 3.2. Many applications can be rephrased as *recovery problems* for functions from unstructured data.

3.2. Generalization

The resulting model function f should be such that it *generalizes* well, *i.e.*, it should give practically useful responses $f(x)$ to new inputs $x \in \Omega$. Furthermore, it should be *stable* in the sense that small changes in the training data do not change f too much. But these goals are in conflict with good reproduction of the training data. A highly stable but useless model would be $f = 1$, while *overfitting* occurs if there is too much emphasis on data reproduction, leading to unstable models with bad generalization properties.

Guideline 3.3. Recovery problems are subject to the *reproduction–generalization dilemma* and need a careful balance between generalization and stability properties on one hand, and data reproduction quality on the other.

This is also called the *bias-variance dilemma* under certain probabilistic hypotheses, but it also occurs in deterministic settings.

Given a recovery problem as in (3.1), there is not enough information to come up with a useful solution of the recovery problem. In particular, we have no idea how to define f or from which space of functions to pick it from. From a theoretical point of view, we are facing an *ill-posed problem* with plenty of indistinguishable approximate solutions. From a practical point of view, all mathematical *a priori* assumptions on f are useless because they do not take the application into account.

Instead, one should use additional application-dependent information concerning the essentials of the inputs, *e.g.*, define a *feature map* $\Phi : \Omega \to \mathcal{F}$

as in (2.2), taking an object x to an object $\Phi(x)$ in $\mathcal{F}$ containing all essential features of x. With this additional information, we can define a kernel K using (2.2), and we get a space $\mathcal{K}$ of functions on Ω via (2.3) and (2.7). Since $\mathcal{K}$ usually turns out to be rather large (see the example in (2.8) for Sobolev spaces), this space serves as a natural reservoir from which to pick f, and if we have no other information, there is no other choice for a space defined on all of Ω. Of course, the choice of a feature map is just another way of adding hypotheses, but it is one that can be tailored perfectly to the application, using kernel engineering knowledge.

3.3. Optimality

We are now left with the problem to pick f somehow from the space $\mathcal{K}$, using our training set. If we insist on exact recovery, we get an instance of Guideline 3.1 from the following theorem.

Theorem 3.4. Let the kernel K be symmetric positive definite. Then a function of the form

$$f^* := \sum_{k=1}^{N} \alpha_k K(\cdot, x_k) \tag{3.4}$$

is the unique minimizer of the Hilbert space norm in $\mathcal{K}$ amongst all functions $f \in \mathcal{K}$ with $f(x_j) = y_j,\ 1 \le j \le N$. The coefficients α_k can be calculated from the linear system

$$\sum_{k=1}^{N} \alpha_k K(x_j, x_k) = y_j, \quad 1 \le j \le N. \tag{3.5}$$

As Section 4 will show, the system (3.5) also arises for different nondeterministic recovery problems in exactly the same way, but with different semantics.

Clearly, symmetric positive definiteness of the kernel implies positive definiteness of the *kernel matrix* $A_{K,X}$ in (3.5) which we saw in Definition 2.6.

Guideline 3.5. Interpolation of unstructured data using a kernel is an optimal strategy for black-box modelling and learning from noiseless information.

The essential information on the application is built into the kernel. Once the kernel is there, things are simple, theoretically. The *generalization error* is optimal in the following sense.

Theorem 3.6. Consider all possible linear recovery schemes of the form

$$f_u(\cdot) := \sum_{j=1}^{N} u_j(\cdot) f(x_j)$$

which use the training data $(x_j, y_j) = (x_j, f(x_j))$, $1 \le j \le N$ for an unknown model $f \in \mathcal{K}$ and employ arbitrary functions u_j on Ω. Then the approximate solution f^* of Theorem 3.4 satisfies

$$\inf_u \sup_{\|f\|_{\mathcal{K}} \le 1} |f(x) - f_u(x)| = \sup_{\|f\|_{\mathcal{K}} \le 1} |f(x) - f^*(x)| \quad \text{for all } x \in \Omega \tag{3.6}$$

and it has the form $f^* = f_{u^*}$ with Lagrange-type functions $u_1^*(x), \dots, u_N^*(x)$ from $\operatorname{span}\{K(\cdot, x_j) \,:\, 1 \le j \le N\}$ satisfying

$$\sum_{j=1}^{N} u_j^*(x) K(x_j, x_k) = K(x, x_k), \quad 1 \le k \le N, \quad \text{for all } x \in \Omega. \tag{3.7}$$

Note that this is another instance of Guideline 3.1. The optimality results of the previous theorems are well-known properties of univariate splines.

Guideline 3.7. In the context of optimal recovery, kernel methods provide natural multivariate extensions of classical univariate spline techniques.

For later reference in Section 4, we should explain the connection between the linear systems (3.5) and (3.7) on one hand, and the representations (3.4) and (3.6) on the other. Theorem 3.4 works on the basis $K(\cdot, x_k)$ directly, while Theorem 3.6 produces a new basis of functions u_j^* which has the Lagrangian property $u_j^*(x_k) = \delta_{jk}$ but spans the same space. The optimal recovery solutions coincide, but have different basis representations.

This basis change, if executed only approximately, is important for applications. In fact, transition to a local Lagrange or 'cardinal' basis is one of the possible preconditioning strategies (Faul and Powell 1999, Ling and Kansa 2004, Brown, Ling, Kansa and Levesley 2005, Ling and Kansa 2005), and approximate Lagrangian bases yield *quasi-interpolants* (Buhmann 1988, Beatson and Light 1993, Buhmann 1993, Buhmann, Dyn and Levin 1995, Maz'ya and Schmidt 2001) which avoid solving linear systems because they provide approximate inverses. This is a promising research area.

3.4. Generalized recovery

If the recovery problem (3.1) is generalized to (3.3), there is a similar theory (Wu 1992, Luo and Levesley 1998) concerning optimal recovery, replacing the kernel matrix with entries $K(x_j, x_k)$ by a symmetric matrix with elements $\lambda_j^x \lambda_k^y K(x, y)$, where we used an upper index x at λ^x to indicate that the functional λ acts with respect to the variable x. The system (3.5) becomes

$$\sum_{k=1}^{N} \alpha_k \lambda_j^x \lambda_k^y K(x, y) = y_j, \quad 1 \le j \le N, \tag{3.8}$$

while (3.7) will turn into

$$\sum_{j=1}^{N} u_j^*(x)\lambda_j^x \lambda_k^y K(x,y) = \lambda_k^y K(x,y), \quad 1 \le k \le N, \quad \text{for all } x \in \Omega.$$

Guideline 3.8. Kernel methods can recover a function f from very general unstructured data, if the kernel is sufficiently smooth and the 'data' of f are linear functionals acting on f.

This is used in applications described in Section 10. In the case of the recovery problem (3.2), we get a symmetric meshless *collocation* technique for solving Poisson's equation. This will be treated in more detail in Section 14.

3.5. Error, condition, and stability

Let us go back to the generalization error. We shall see in Section 7 that the generalization error of kernels on $\mathbb{R}^d$ dramatically improves with their smoothness while still maintaining applicability to recovery problems with unstructured data. This is one of the key features of kernel techniques.

Guideline 3.9. Methods based on fixed smooth positive definite kernels can provide recovery techniques with very small errors, using rather small amounts of data.

But the small generalization error comes at a high price, because there are serious practical problems with systems of the form (3.5). This is in sharp contrast to the encouraging optimality properties stated so far.

Guideline 3.10. The linear systems (3.5) and (3.8) can be very large, non-sparse and severely ill-conditioned.

However, the latter is no surprise because the method solves an ill-posed problem approximately. Thus the bad condition of the system (3.5) must be expected somehow. There is an apparent link between condition and scaling, since kernels with small supports will lead to approximately diagonal kernel matrices, while kernels with wide scales produce matrices with very similar rows and columns.

Guideline 3.11. Positive definite kernels with small scales lead to better matrix condition than kernels with wide scales.

Since we know that kernel systems (3.5) or (3.8) are solvable for symmetric positive definite kernels and linearly independent data functionals, we have the following guideline.

Guideline 3.12. Methods based on positive definite kernels have a built-in *regularization.*

In fact, they solve the ill-posed problem (3.1) by providing an approximate solution minimizing the Hilbert space norm in $\mathcal{K}$ under all conceivable exact recovery schemes, as if they were using a regularizing penalty term of the form $\|f\|_{\mathcal{K}}^2$, which can be a Sobolev space norm for certain kernels. This regularization property will come up later when we use kernels in collocation techniques for solving partial differential equations. If (3.5) is viewed as an approximate solution of the integral equation

$$\int_\Omega \alpha(x)K(y,x)\,\mathrm{d}x = f(y) \quad \text{for all } y \in \Omega$$

via a quadrature formula, we have another aspect telling us that (3.5) solves an ill-posed problem approximately via some regularization in the background. Note the connection to convolution (2.4).

The generalization error $f(x) - f^*(x)$ and the condition of the system (3.5) have an unexpected connection. Theoretical results (Schaback 1995*a*) and simple numerical experiments with various kernels show the following.

Guideline 3.13. Increasing smoothness of kernels on $\mathbb{R}^d$ decreases the recovery error but increases the condition of the system (3.5). There are no kernels that provide small errors and good condition simultaneously.

Guideline 3.14. Increasing the scale of a kernel on $\mathbb{R}^d$ decreases the recovery error but increases the condition of the system (3.5).

Note that this limits the use of systems like (3.5) in their original form, but techniques like preconditioning (Faul and Powell 1999, Ling and Kansa 2004, Brown *et al.* 2005, Ling and Kansa 2005) or domain decomposition (see Section 8) should be applied.

Guideline 3.15. Programming of kernel methods should always use the kernel scaling as an adjustable parameter. Experiments with different scales often show that systems without preconditioning give best results when the scaling is as wide as numerically feasible. Following Guideline 3.17 below, one should use pivoting or SVD techniques, and this can work well beyond the standard condition limit of 10^{15}.

Reasons for this will be given at the end of this section. The limit behaviour of recovery problems for analytic kernels with increasing width is related to multivariate polynomials (Driscoll and Fornberg 2002). Unexpectedly, function recovery problems using wide-scaled Gaussians tend to the polynomial interpolation method of de Boor and Ron, if the scales get infinitely wide (Schaback 2005*a*). This will hopefully lead to a better understanding of preconditioning techniques in the future.

3.6. Relaxation and complexity

If N is huge, the exact solution (3.4) of a system (3.5) is much too complex to be useful. This is where another general rule comes up.

Guideline 3.16. Within kernel methods, relaxation of requirements can lead to reduction of complexity.

Under certain probabilistic hypotheses, this is another aspect of the *bias-variance dilemma* related to *overfitting*. As we mentioned at the beginning, insisting on exact reproduction of huge amounts of data increases the complexity of the model and makes it very sensible to changes in the training data, thus less reliable as a model. Conversely, relaxing the reproduction quality will allow a simpler model. Before we turn to specific relaxation methods used in kernel-based learning, we should look back at Guideline 3.9 to see that badly conditioned large systems of the form (3.5) using smooth kernels will often have subsystems that provide good approximate solutions to the full system. This occurs if the generalization error is small when going over from the training data of a subset to the full training data. Thus Guideline 3.16 can be satisfied by simply taking a small suitable subset of the data, relying on Guideline 3.9. As we shall see, this has serious implications for kernel-based techniques for solving partial differential equations or machine learning. For simple cases, the following suffices.

Guideline 3.17. Within kernel methods, large and ill-conditioned systems often have small and better conditioned subsystems furnishing good approximate solutions to the full system. Handling numerical rank loss by intelligent pivoting is useful.

However, large problems need special treatment, and we shall deal with such cases in Section 8.

The relaxation of (3.5) towards (3.1) can be done in several ways, and learning theory uses *loss functions* to quantify the admissible error in (3.1). We present this in Section 11 in more detail. Let us look at a simple special case. We allow a uniform tolerance ϵ on the reproduction of the training data, *i.e.*, we impose the linear constraints

$$-\epsilon \leq f(x_j) - y_j \leq \epsilon, \quad 1 \leq j \leq N. \tag{3.9}$$

We then minimize $\|f\|_{\mathcal{K}}$ while keeping ϵ fixed, or we minimize the weighted objective function $\frac{1}{2}\|f\|_{\mathcal{K}}^2 + C\epsilon$ when ϵ is varying and C is fixed. Optimization theory then tells us that the solution f^* is again of the form (3.4), but the Kuhn–Tucker conditions imply that the sum only contains terms where the constraints in (3.9) are *active*, *i.e.*, $\alpha_k \neq 0$ holds only for those k with $|f(x_k) - y_k| = \epsilon$. In view of principle (3.9) these *support vectors* will often be a rather small subset of the full data, and they provide an instance of complexity reduction via relaxation according to Guideline 3.16. This roughly describes the principles behind *support vector machines* for the implementation of learning algorithms. These principles are consequences of optimization, not of statistical learning theory, and they arise in other applications as well. We explain this in some more detail in Section 11 and apply it to adaptive collocation solvers for partial differential equations in Section 14.

Furthermore, we see via this optimization argument that the exact solution of a large system (3.5) can be replaced by an approximate solution of a smaller subsystem. This supports Guideline 3.17 again. It is in sharpest possible contrast to the large linear systems arising in finite element theory.

Guideline 3.18. Systems arising in kernel-based recovery problems should be solved approximately by adaptive or optimization algorithms, finding suitable subproblems.

At this point, the idea of *online learning* is helpful. It means that the training sample is viewed as a possibly infinite input sequence $(x_j, y_j) \approx (x_j, f(x_j)),\ j = 1, 2, \dots$ which is used to update the current model function f_k if necessary. The connection to adaptive recovery algorithms is clear, since a new training data pair (x_{N+1}, y_{N+1}) will be discarded if the current model function f_k works well on it, *i.e.*, if $f_k(x_{N+1}) - y_{N+1}$ is small. Otherwise, the model function is carefully and efficiently updated to make optimal use of the new data (Schaback and Werner 2006). Along these lines, one can devise adaptive methods for the approximate solution of partial differential equations which 'learn' the solution in the sense of online learning, if they are given infinitely many data of the form (3.2).

Within approximation theory, the concept of adaptivity is closely related to the use of *dictionaries* and *frames*. In both cases, the user does not work with a finite and small set of trial functions to perform a recovery. Instead, a selection from a large reservoir of possible trial functions is made, *e.g.*, by *greedy* adaptive methods or by choosing frame representations with many vanishing coefficients via certain projections. This will be a promising research area in the coming years.

The final sections of this article will review several application areas of kernel techniques. However, we shall follow the principles stated above, and we shall work out the connections between recovery, learning, and equation solving at various places. This will have to start with a look at nondeterministic recovery problems.

4. Kernels in probabilistic models

There are several different ways in which kernels arise in probability theory and statistics. We shall describe the most important ones very briefly, ignoring the standard occurrence of certain kernels like the Gaussian as *densities* of probability distributions. Since *Acta Numerica* is aiming at readers in numerical analysis, we want to assume as little stochastic background as possible.

4.1. Nondeterministic recovery problems

If we go back to the recovery problem of Section 3 and rewrite it in a natural probabilistic setting, we get another instance of Guideline 3.1, because kernel-based techniques again turn out to have important optimality properties. Like in Section 3 we assume that we want to find the response $f(x)$ of an unknown model function f at a new point x of a set Ω, provided that we have a sample of input-response pairs $(x_j, y_j) = (x_j, f(x_j))$ given by observation or experiment. But now we assume that the whole setting is nondeterministic, *i.e.*, the response y_j at x_j is not a fixed function of x_j but rather a realization of a real-valued random variable $Z(x_j)$. Thus we assume that for each $x \in \Omega$ there is a real-valued random variable $Z(x)$ with expectation $E(Z(x))$ and bounded positive variance $E((Z(x) - E(Z(x))^2)$. The goal is to get information about the function $E(Z(x))$ which replaces our f in the deterministic setting. For two elements $x, y \in \Omega$ the random variables $Z(x)$ and $Z(y)$ will not be uncorrelated, because if x is close to y the random experiments described by $Z(x)$ and $Z(y)$ will often show similar behaviour. This is described by a *covariance kernel*

$$\operatorname{cov}(x, y) := E(Z(x) \cdot Z(y)) \quad \text{for all } x, y \in \Omega. \tag{4.1}$$

Such a kernel exists and is positive semidefinite under weak additional assumptions. If there are no exact linear dependencies in the random variables $Z(x_i)$, a kernel matrix with entries $\operatorname{cov}(x_j, x_k)$ will be positive definite. A special case is a *Gaussian process* on Ω, where for every subset $X = \{x_1, \ldots, x_N\} \subset \Omega$ the vectors $Z_X := (Z(x_1), \ldots, Z(x_N))$ have a multivariate Gaussian distribution with mean $E(Z_X) \in \mathbb{R}^N$ and a covariance yielding a matrix $A \in \mathbb{R}^{N \times N}$ which has entries $\operatorname{cov}(x_j, x_k)$ in the above sense. Note that this takes us back to the *kernel matrix* of Definition 2.6 and the system (3.5).

Now there are several equivalent approaches to produce a good estimate for $Z(x)$ once we know data pairs (x_j, y_j) where the y_j are noiseless realizations of $Z(x_j)$. The case of additional noise will be treated later.

First, *Bayesian* thinking asks for the expectation of $Z(x)$ given the information $Z(x_1) = y_1, \ldots, Z(x_N) = y_N$ and write this as the expectation of a conditional probability

$$\tilde{Z}(x) := E(Z(x) | Z(x_1) = y_1, \ldots, Z(x_N) = y_N).$$

This is a function of x and all data pairs, and it serves as an approximation of $E(Z(x))$.

Second, *estimation theory* looks at all linear estimators of the form

$$\tilde{Z}(x) := \sum_{j=1}^{N} u_j(x) y_j$$

using the known data to predict $Z(x)$ optimally. It minimizes the *risk* defined as

$$E\left(\left(Z(x) - \sum_{j=1}^{N} u_j(x) Z(x_j)\right)^2\right)$$

by choosing appropriate coefficients $u_j(x)$.

Both approaches give the same result, repeating Theorems 3.4 and 3.6 with a new probabilistic interpretation. Furthermore, the result is computationally identical to the solution of the deterministic case using the kernel $K(x, y) = \operatorname{cov}(x, y)$ right away, ignoring the probabilistic background completely. The system (3.5) has to be solved for the coefficients α_k, and the result can be written via either (3.4) or Theorem 3.6. The proof of this theorem is roughly the same as that for the estimation theory case in the probabilistic setting.

Guideline 4.1. Positive definite kernels allow a unified treatment of deterministic and probabilistic methods for recovery of functions from data.

Guideline 4.2. Applications using kernel-based trial spaces in non-deterministic settings should keep in mind that what they do is equivalent to an estimation process for spatial random variables with a covariance described by the chosen kernel.

This means that compactly supported or quickly decaying kernels lead to uncoupled spatial variables at larger distances. Furthermore, it explains why wide scales usually allow us to get along with fewer data (see Guideline 3.14). If there is a strong interdependence of local data, it suffices to use few data to explain the phenomena.

If the covariance kernel is positive definite, the general theory of Section 2 applies. It turns the space spanned by functions $\operatorname{cov}(\cdot, y)$ on Ω into a reproducing kernel Hilbert space such that the inner product of two such functions is expressible via (2.5) by the covariance kernel itself. This is not directly apparent from where we started. In view of learning theory, the map $x \mapsto \operatorname{cov}(x, y)$ is a special kind of feature map which assigns to each other input x a number indicating how closely related it is to y.

4.2. Noisy data

If we add a noise variable $\epsilon(x)$ at each point $x \in \Omega$ with mean zero and variance σ^2 such that the noise at different points is independent and also independent of Z, the covariance kernel with noise is

$$E((Z(x)+\epsilon(x))\cdot(Z(y)+\epsilon(y))) = \operatorname{cov}(x,y)+\sigma^2\delta_{xy}.$$

Thus, in the presence of noise we have to add a diagonal matrix with entries σ^2 to the kernel matrix in (3.5). This addition of noise makes the kernel matrices positive definite even if the covariance kernel is only positive semidefinite. In a deterministic setting, this reappears as *relaxed interpolation* and will be treated in Section 7.

If there is no *a priori* information on the covariance kernel and the noise variance σ, one can try to estimate these from a sufficiently large data sample. For details we refer to the vast statistical literature concerning noise estimation and techniques like cross-validation. Choosing the relaxation parameter in the deterministic case will be treated in some detail in Section 7, with references given there.

4.3. Random functions

In the above situation we had a random variable $Z(x)$ at each point $x \in \Omega$. But one can also consider random choices of functions f from a set or space $\mathcal{F}$ of real-valued functions on Ω. This requires a probability measure ρ on $\mathcal{F}$, and one can define another kind of *covariance kernel* via

$$\begin{aligned}\operatorname{cov}(x,y) &:= E(f(x)\cdot f(y))\\ &= \int_{\mathcal{F}} f(x)f(y)\,\mathrm{d}\rho(f) \qquad \text{for all } x,y\in\Omega \qquad (4.2)\\ &= \int_{\mathcal{F}} \delta_x(f)\delta_y(f)\,\mathrm{d}\rho(f) \quad \text{for all } x,y\in\Omega.\end{aligned}$$

This is a completely different situation, both mathematically and 'experimentally', because the random events and probability spaces are different.

But now the connection to Hilbert spaces and feature maps is much clearer right from the start, since the final form of the covariance kernel can be seen as a bilinear form $\operatorname{cov}(x,y) = (\delta_x,\delta_y)$ in a suitable space. For this, we define a feature map

$$\Phi(x) := \delta_x \;:\; f \mapsto f(x) \quad \text{for all } f\in\mathcal{F} \tag{4.3}$$

as a linear functional on $\mathcal{F}$. To a fixed input item x it assigns all possible 'attributes' or 'features' $f(x)$ where f varies over all random functions in $\mathcal{F}$. If we further assume that the range of the feature map is a pre-Hilbert subspace of the dual $\mathcal{F}^*$ of $\mathcal{F}$ under the inner product

$$(\lambda,\mu)_{\mathcal{F}^*} := E(\lambda(f)\cdot\mu(f)) = \int_{\mathcal{F}} \lambda(f)\mu(f)\,\mathrm{d}\rho(f),$$

we are back to (2.2) in the form

$$\mathrm{cov}(x,y) = (\Phi(x), \Phi(y))_{\mathcal{H}} \quad \text{for all } x, y \in \Omega \tag{4.4}$$

once we take $\mathcal{H}$ as the Hilbert space completion.

If we have training data pairs (x_i, y_i), $i = 1, \ldots, N$ as before, the y_i are simultaneous evaluations $y_i = f(x_i)$ of a random function $f \in \mathcal{F}$. A Bayesian recovery problem without noise would take the expected $f \in \mathcal{F}$ under the known information $y_i = f(x_i)$ for $i = 1, \ldots, N$. Another approach is to find functions u_j on Ω such that the expectation

$$E\left(\left(f(x) - \sum_{j=1}^{N} u_j(x) f(x_j)\right)^2\right)$$

is minimized. Again, these two recovery problems coincide and are computationally equivalent to those treated in Section 2 in the deterministic case, once the covariance kernel is specified.

The two different definitions for a covariance kernel cannot lead to serious confusion, because they are very closely related. If we start with random functions and (4.2), there are pointwise random variables $Z(x) := \{f(x)\}_{f \in \mathcal{F}}$ leading to the same covariance kernel via (4.1). Conversely, starting from random variables $Z(x)$ and (4.1) such that the covariance kernel is positive definite, a suitable function class $\mathcal{F}$ can be defined via the span of all $\mathrm{cov}(\cdot, y)$, and point evaluations on this function class carry an inner product which allows us to define a Hilbert space $\mathcal{H}$ such that (4.3) and (4.4) hold.

From here on, statistical learning theory (Schölkopf and Smola 2002, Shawe-Taylor and Cristianini 2004) takes over, and we refer to the two cited books.

4.4. Density estimation by kernels

This is again a different story, because the standard approach does not solve a linear system. The problem is to recover the density f of a multivariate distribution over a domain Ω from a large sample $x_1, \ldots, x_N \in \Omega$ including repetitions. The true density function must take large values in regions where the density of sampling points is high. A primitive density estimate is possible via counting the samples in each cell of a grid, and to plot the resulting histogram. This yields a piecewise constant density estimate, but we can do better by using a nonnegative symmetric translation-invariant kernel K with total integral one, and defining

$$\tilde{f}(x) := \tfrac{1}{N} \sum_{j=1}^{N} K\left(\tfrac{x - x_i}{h}\right)$$

as a smooth estimator. If the *bandwidth* h is taken too small, the result just shows sharp peaks at the x_i. If h is too large, the result is smoothed too much to be useful. We have another instance of the *scaling problem* here. Statisticians have quite some literature on picking the 'right' bandwidth and kernel experimentally, using as much observational or *a priori* information as possible, but we cannot deal with these here.

5. Kernel construction

Before we delve into applications, we have to prepare by taking a closer and more application-oriented view at kernels. We want to give a short but comprehensive account of kernel construction techniques, making the reader able to assess features of given kernels or to construct new ones with prescribed properties.

If the domain Ω has no structure at all, we already know that the most important strategy to get a useful kernel is to construct a feature map $\Phi\ : \Omega \to \mathcal{F}$ with values in some Hilbert space $\mathcal{F}$ first, and then to use (2.2) for definition of a kernel. The resulting kernel is always positive semidefinite, but it will be hard to check for positive definiteness *a priori*, because this amounts to proving that, for arbitrary different $x_j \in \Omega$, the feature vectors $\Phi(x_j) \in \mathcal{F}$ are linearly independent. However, linearly dependent $\Phi(x_j)$ lead to linearly dependent functions $K(\cdot, x_j)$, and these are useless in the representation (3.4) and can be blended out by pivoting or a suitable optimization.

Guideline 5.1. If pivoting, adaptivity, or optimization is used according to Guidelines 3.17 and 3.18, one can safely work with positive *semi*definite kernels in practice.

5.1. Mercer kernels

A very common special case of a feature map occurs if there is a finite or countable set $\{\varphi_i\}_{i\in I}$ of functions on Ω. In applications, this arises if $\varphi_i(x)$ is the value of feature number i on an element $x \in \Omega$. The feature map Φ then takes an element x into the set $\Phi(x) := \{\varphi_i(x)\}_{i\in I} \in \mathbb{R}^{|I|}$. For a set $\{w_i\}_{i\in I}$ of positive weights one can define a weighted ℓ_2 space by

$$\ell_{2,w}(I) := \left\{ \{c_i\}_{i\in I}\ :\ \sum_{i\in I} w_i c_i^2 < \infty \right\}$$

and then assume that these weights and the functions φ_i satisfy

$$\sum_{i\in I} w_i \varphi_i^2(x) < \infty$$

on all of Ω. This means that the scaling of the functions φ_i together with the weights w_i must be properly chosen such that the above series converges. Then we define $\mathcal{F} := \ell_{2,w}(I)$ and (2.2) yields the kernel

$$K(x,y) := \sum_{i\in I} w_i\varphi_i(x)\varphi_i(y) \quad \text{for all } x,y\in\Omega \tag{5.1}$$

dating back to early work of Hilbert and Schmidt. Such kernels are called *Mercer* kernels in the context of learning algorithms because of their connection to the Mercer theorem on positive integral operators. But note that the latter theory is much more restrictive, decomposing a given positive integral operator with kernel K into orthogonal eigenfunctions φ_i corresponding to eigenvalues w_i. For our purposes, such assumptions are not necessary.

Even outside machine learning, many useful recovery algorithms use kernels of the above form. For instance, on spheres one can take spherical harmonics, and on tori one can take *sin* and *cos* functions as the φ_i. This is the standard way of handling kernels in these situations, and there is a huge literature on such methods, including applications to geophysics. We describe the case of the sphere in Section 9 and provide references there.

The reader may figure out that finite partial sums of (5.1) are well-known ingredients of calculus. For instance, classical Fourier analysis on $[0,2\pi)$ or the unit circle in the complex plane using standard trigonometric functions and fixed weights leads to the well-known *Dirichlet* kernel this way. If the functions φ_i are orthogonal univariate polynomials, the corresponding kernel is provided by the *Christoffel–Darboux* formula.

Guideline 5.2. If expansion-type kernels (5.1) are used, kernel methods provide natural multivariate extensions not only of splines (see Guideline 3.7), but also of classical univariate techniques based on orthogonality.

A highly interesting new class of kernels arises when the functions φ_i are scaled shifts of compactly supported refinable functions in the sense of wavelet theory. The resulting *multiscale kernels* (Opfer 2006) have a built-in multiresolution structure relating them to wavelets and frames. Implementing these new kernels into known kernel techniques yields interesting multiscale algorithms which are currently investigated.

5.2. Convolution kernels

Of course, one can generalize (5.1) to a convolution-type formula, *i.e.*,

$$K(x,y) := \int_T \varphi(x,t)\varphi(y,t)w(t)\,\mathrm{d}t \quad \text{for all } x,y\in\Omega \tag{5.2}$$

under certain integrability conditions and with a positive weight function w on an integration domain T. This always yields a positive semidefinite kernel, and positive definiteness follows if functions $\varphi(x,\cdot)$ are linearly independent on T for different $x \in \Omega$ (Levesley, Xu, Light and Cheney 1996). Together with (5.1), this technique is able to generate compactly supported kernels, but there are no useful special cases known which were constructed along these lines.

Guideline 5.3. Kernels obtained by weighted positive summation or by convolution of products of other functions are positive semidefinite.

5.3. Kernels and harmonic analysis

However, the most important case arises when the underlying set Ω has more structure, in particular if it is a locally compact abelian group and allows *transforms* of some sort.

Guideline 5.4. Invariant kernels with positive transforms are positive semidefinite.

We do not want to underpin this in full generality, *e.g.*, for Riemannian manifolds (Dyn, Narcowich and Ward 1999) or for topological groups (Gutzmer 1996, Levesley and Kushpel 1999). Instead, we focus on translation-invariant kernels on $\mathbb{R}^d$ and use Fourier transforms there, where the above result is well known and easy to prove. In fact, positivity of the Fourier transform almost everywhere is sufficient for positive definiteness of a kernel. This argument proves positive definiteness of the Gaussian and the Sobolev kernel in (2.8), because their Fourier transforms are well known (another Gaussian and the function $(1+\|\cdot\|_2^2)^{-k}$, respectively, up to certain constants). By inverse argumentation, the *inverse multiquadric* kernels of the form

$$K(\|x-y\|_2) := (1+\|x-y\|_2^2)^{-k}, \quad x,y \in \mathbb{R}^d,\ k > d/2 \tag{5.3}$$

are also positive definite.

5.4. Compactly supported kernels

But note that all of these kernels have infinite support, and the kernel matrices arising in (3.5) will not be sparse. To generate sparse kernel matrices, we need explicitly known compactly supported kernels with positive Fourier transforms. This was quite a challenge for some years, but now there are classes of such kernels explicitly available via efficient representations (Wu 1995, Wendland 1995, Buhmann 1998). If they are dilated to have support on the unit ball, they have the simple radial form

$$K(x,y) = \phi(\|x-y\|_2) = \begin{cases} p(\|x-y\|_2), & \text{if } \|x-y\|_2 \le 1, \\ 0, & \text{else,} \end{cases} \tag{5.4}$$

Table 5.1. Wendland's functions $\phi_{d,k}$.

Space dimension	Function	Smoothness
$d = 1$	$\phi_{1,0}(r) = (1-r)_+$	C^0
	$\phi_{1,1}(r) \doteq (1-r)_+^3(3r+1)$	C^2
	$\phi_{1,2}(r) \doteq (1-r)_+^5(8r^2+5r+1)$	C^4
$d \leq 3$	$\phi_{3,0}(r) = (1-r)_+^2$	C^0
	$\phi_{3,1}(r) \doteq (1-r)_+^4(4r+1)$	C^2
	$\phi_{3,2}(r) \doteq (1-r)_+^6(35r^2+18r+3)$	C^4
	$\phi_{3,3}(r) \doteq (1-r)_+^8(32r^3+25r^2+8r+1)$	C^6
$d \leq 5$	$\phi_{5,0}(r) = (1-r)_+^3$	C^0
	$\phi_{5,1}(r) \doteq (1-r)_+^5(5r+1)$	C^2
	$\phi_{5,2}(r) \doteq (1-r)_+^7(16r^2+7r+1)$	C^4

where p is a univariate polynomial in the case of Wu's and Wendland's functions. For Buhmann's functions, p contains an additional log-factor. In particular, Wendland's functions are well studied for various reasons. First of all, given a space dimension d and degree of smoothness $2k$, the polynomial $p = \phi_{d,k}$ in (5.4) has minimal degree among all positive definite functions of the form (5.4). Furthermore, their 'native' reproducing kernel Hilbert spaces are norm-equivalent to Sobolev spaces $H^\tau(\mathbb{R}^d)$ of order $\tau = d/2 + k + 1/2$. Finally, the simple structure allows a fast evaluation. Examples of these functions are given in Table 5.1.

5.5. Further cases

There are a few other construction techniques that allow us to generate new kernels out of known ones.

Theorem 5.5. Kernels obtained by weighted positive summation of positive (semi-) definite kernels on the same domain Ω are positive (semi-) definite.

For handling data involving differential operators, we need the following.

Guideline 5.6. If a nontrivial linear operator L is applied to both arguments of a positive semidefinite kernel, then it will in most cases be possible to construct another positive semidefinite kernel.

This can be carried out in detail by using the representations (5.1) or (5.2), if they are available. In general, one can work with (2.5) and assume that L can be applied inside the inner product.

There is a final construction technique we only mention here briefly. It is covered well in the literature, dating back to Hausdorff, Bernstein and Widder, and it was connected to completely monotone univariate functions by Schoenberg and Micchelli (Micchelli 1986). It is of minor importance for constructing application-oriented kernels, because it is restricted to radial kernels which are positive definite on $\mathbb{R}^d$ for all dimensions, and it cannot generate kernels with compact support. However, it provides useful theoretical tools for analysing the kernels which follow next.

6. Special kernels

So far, we have already presented the Gaussian kernel (2.1), the inverse multiquadric (5.3), and the Sobolev kernel (2.8). These have in common that they are *radial basis functions* which are globally positive and have positive Fourier transforms. Another important class of radial kernels is compactly supported and of local polynomial form, *i.e.*, the Wendland functions (5.4). But this is not the end of all possibilities.

6.1. Kernels as fundamental solutions

Guideline 6.1. There are other and somewhat more special kernels which are related to important partial differential equations.

The most prominent case is the *thin-plate spline* (Duchon 1976, 1979)

$$K(x,y) = \|x-y\|_2^2 \log \|x-y\|_2 \quad \text{for all } x,y \in \mathbb{R}^d, \tag{6.1}$$

which models a thin elastic sheet suspended at y as a function of x and solves the biharmonic equation $\Delta^2 u = 0$ in two dimensions, everywhere except at y. More generally, there are *polyharmonic splines* defined as fundamental solutions of iterated Laplacians. They deserve a closer look, because they have special scaling properties, are of central importance for the meshless *method of fundamental solutions* in Section 13, and lead naturally to the notion of *conditionally positive definite functions* below.

The *fundamental solution* for a differential operator L at some point $x \in \mathbb{R}^d$ is defined as a kernel $K(x,\cdot)$ which satisfies $LK(x,\cdot) = \delta_x$ in the distributional sense. For the iterated Laplacian $L_m := (-\Delta)^m$ we get radial kernels

$$\begin{aligned} &r^{2m-d} && \text{for } d \text{ odd},\\ &r^{2m-d}\log r && \text{for } d \text{ even}, \end{aligned}$$

as functions of $r = \|x-y\|_2$ up to multiplicative constants and for $2m > d$. This contains the thin-plate splines of (6.1) for $m = d = 2$ and generalizes to positive real exponents as

$$\begin{aligned} r^\beta &\quad \text{for } \beta \notin 2\mathbb{Z}, \\ r^\beta \log r &\quad \text{for } \beta \in 2\mathbb{Z}, \end{aligned} \tag{6.2}$$

where now the space dimension no longer appears.

Unfortunately, these functions increase with r, and so they are neither bell-shaped nor globally integrable. Their Fourier transforms cannot be calculated in the classical sense, and thus there are no standard Fourier transform techniques to prove positive definiteness. The same holds for *multiquadrics*

$$(1+r^2)^{\beta/2} \quad \text{for } \beta \notin 2\mathbb{Z},\ \beta > 0,$$

which can be seen as a regularization of the polyharmonic spline r^β at zero, and which extends the inverse multiquadrics of (5.3) to positive exponents, the most widely used case being $\beta = 1$. Fortunately, these functions can be included within kernel theory by a simple generalization.

6.2. Conditionally positive definite kernels

Definition 6.2. A symmetric kernel $K : \Omega \times \Omega \to \mathbb{R}$ is *conditionally positive* (*semi-*) *definite* of order m on $\Omega \subseteq \mathbb{R}^d$, if for all finite subsets $X := \{x_1, \dots, x_N\}$ of distinct points in Ω the symmetric matrices $A_{K,X}$ with entries $K(x_j, x_k)$, $1 \le j, k \le N$ define a positive (semi-) definite quadratic form on the subspace

$$V_{m,X} := \left\{ \alpha \in \mathbb{R}^N \ : \ \sum_{j=1}^N \alpha_j p(x_j) = 0 \quad \text{for all } p \in \pi_{m-1}(\mathbb{R}^d) \right\} \tag{6.3}$$

of coefficient vectors satisfying certain ‘discrete vanishing moment’ conditions with respect to the space $\pi_{m-1}(\mathbb{R}^d)$ of d-variate real polynomials of degree smaller than m.

Note that (unconditional) positive definiteness is identical to conditional positive definiteness of order zero, and that conditional positive definiteness of order m implies conditional positive definiteness of any larger order. Table 6.1 lists the appropriate orders of positive definiteness for special radial kernels.

Recovery problems using conditionally positive definite kernels of positive order m have to modify the trial space $\mathcal{K}_0$ to

$$\begin{aligned} \mathcal{K}_m &:= \pi_{m-1}(\mathbb{R}^d) + \mathcal{P}_m, \\ \mathcal{P}_m &:= \operatorname{span}\left\{ \sum_{j=1}^N \alpha_j K(\cdot, x_j),\ \alpha \in V_{m,X},\ X = \{x_1, \dots, x_N\} \subset \Omega \right\}. \end{aligned} \tag{6.4}$$

The norm (2.6) now works only on the space $\mathcal{P}_m$, and thus only $\operatorname{clos}\mathcal{P}_m$ turns into a Hilbert space. The native space $\mathcal{K}$ for K is then

$$\mathcal{K} := \pi_{m-1}(\mathbb{R}^d) + \operatorname{clos}\mathcal{P}_m,$$

but the reproduction (2.9) of functions via the kernel K needs a modification which we shall omit here.

If we have training data (x_k, y_k), $1 \le k \le N$ for a model $f(x_k) = y_k$, we now plug these equations into our new trial space, using a basis $p_1, \ldots, p_Q$ of $\pi_{m-1}(\mathbb{R}^d)$ and get a linear system

$$\begin{array}{rcrcll} \displaystyle\sum_{j=1}^{N} \alpha_j K(x_k, x_j) & + & \displaystyle\sum_{i=1}^{Q} \beta_i p(x_k) & = & y_k, & 1 \le k \le N, \\ \displaystyle\sum_{j=1}^{N} \alpha_j p_\ell(x_j) & + & 0 & = & 0, & 1 \le \ell \le Q. \end{array}$$

This system has $N+Q$ equations and unknowns, and it is uniquely solvable if there is no nonzero polynomial vanishing on the set $X = \{x_1, \ldots, x_N\}$. Since the order m of conditional positive definiteness is usually rather small ($m = 1$ for standard multiquadrics and $K(x, y) = \|x - y\|_2$, while $m = 2$ for thin-plate splines) this modification is not serious, and it can be made obsolete if the kernel is changed slightly (Light and Wayne 1998, Schaback 1999). However, many engineering applications use multiquadrics or thin-plate splines without adding constant or linear polynomials, and without caring for the moment conditions in (6.3). This often causes no visible problems, but violates restrictions imposed by conditional positive definiteness.

Note that trial spaces for polyharmonic functions are independent of scaling, if they are properly defined via (6.4). This eliminates many of the scaling problems arising in applications, but it comes at the price of limited smoothness of the kernels, thus reducing the attainable reproduction errors according to Guideline 3.13.

Table 6.1. Orders of conditional positive definiteness.

Kernel $\Phi(r)$, $r = \|x - y\|_2$	Order m	Conditions
$(-1)^{\lceil \beta/2 \rceil}(c^2 + r^2)^{\beta/2}$	$\lceil \beta/2 \rceil$	$\beta > 0$, $\beta \notin 2\mathbb{N}$
$(-1)^{\lceil \beta/2 \rceil} r^\beta$	$\lceil \beta/2 \rceil$	$\beta > 0$, $\beta \notin 2\mathbb{N}$
$(-1)^{k+1} r^{2k} \log r$	$k+1$	$k \in \mathbb{N}$

6.3. Singular kernels

The condition $2m > d$ for the polyharmonic functions forbids useful cases like $m = 1$ in dimensions $d \geq 2$, and thus it excludes the fundamental solutions $\log r$ and r^{-1} of the Laplacian in dimensions 2 and 3. These kernels are radial, but they have singularities at zero. They still are useful reproducing kernels in Sobolev spaces $W_2^1(\mathbb{R}^d)$ for $d = 2, 3$, but the reproduction property now reads

$$\lambda(f) = (\lambda^x K(\cdot - x), f)_{W_2^1(\mathbb{R}^d)} \tag{6.5}$$

for all $f \in W_2^1(\mathbb{R}^d)$, $\lambda \in \left(W_2^1(\mathbb{R}^d)\right)^* = W_2^{-1}(\mathbb{R}^d)$. These kernels and their derivatives arise in integral equations as *single* or *double layer potentials*, and we shall encounter them again in Section 13 where they are used for the meshless *method of fundamental solutions*.

7. Approximation by kernels

This section serves to support Guideline 3.9 concerning the surprising quality of kernel-based approximations. We shall do this in a strictly deterministic setting, ignoring, for instance, the interesting results from statistical learning theory.

7.1. Convolution approximation

One of the oldest forms of kernel approximation is used for *series expansions* and *mollifiers*, and it takes the form of *convolution*. It is also at the core of *smoothed particle hydrodynamics*, a class of practically very useful meshless kernel-based methods we briefly describe in Section 12. Here, we use it as an introduction to the behaviour of kernel approximations in general.

Global convolution of a given function f with a kernel K is

$$K * f := \int_{\mathbb{R}^d} f(x) K(\cdot - x)\, \mathrm{d}x,$$

where we have restricted ourselves to translation-invariant kernels on $\mathbb{R}^d$. Approximation of a function f by kernel convolution means

$$f \approx K * f \tag{7.1}$$

in some norm. Clearly, equality in (7.1) holds only if the kernel acts like a delta functional. Thus convolutions with kernels should achieve good reproduction if the kernels are approximations to the delta functional. This indicates that *scaling* is a crucial issue here again. If K is smoother than f, convolution allows us to construct smooth approximations to nonsmooth functions.

To deal with scaling properly, we observe Guideline 3.15 and introduce a positive scaling parameter δ to scale a fixed kernel K in $L_1(\mathbb{R}^d)$ by

$$K_\delta(\cdot) := \delta^{-d} K(\cdot/\delta)$$

to make the integral of K_δ on $\mathbb{R}^d$ independent of δ. Furthermore, the kernel convolution should approximate monomials $p_\alpha(x) := x^\alpha$ of order at most k well in a *pointwise* sense, *i.e.*, for all $|\alpha| < k,\ \delta > 0$ we require

$$|K_\delta * p_\alpha - p_\alpha|(x) \le \delta^k A(x) \quad \text{for all } x \in \mathbb{R}^d \tag{7.2}$$

with a fixed function A on $\mathbb{R}^d$. For some positive integer k we finally assume that the kernel satisfies a *decay* condition

$$p_\alpha \cdot K \in L_1(\mathbb{R}^d) \quad \text{for all } |\alpha| < k. \tag{7.3}$$

Theorem 7.1. (Cheney, Light and Xu 1992, Cheney and Light 2000) Under these assumptions, there is a constant c such that

$$\|K_\delta * f - f\|_{L_\infty(\mathbb{R}^d)} \le c\delta^k \max_{|\alpha|\le k} \|f^\alpha\|_{L_\infty(\mathbb{R}^d)} \tag{7.4}$$

holds for all functions $f \in C^k(\mathbb{R}^d)$.

Note that the convergence depends on the scale parameter δ going to zero, while the rate is dependent on the decay of K. Surprisingly, the reproduction condition (7.2) can always be achieved exactly (Cheney and Light 2000) by a suitable linear combination of scaled instances of the original kernel, provided that it satisfies the decay condition (7.3) and has integral one. However, this modification of the kernel will in general spoil positive definiteness. Similar kernel modifications arise in many application-oriented papers on meshless kernel methods.

7.2. Discretized convolution approximation

Discretization of the convolution integral leads to

$$(K_\delta * f)(x) \approx \sum_{i \in I_\delta} f(x_{i,\delta}) K_\delta(x - x_{i,\delta}) w_{i,\delta} \quad \text{for all } x \in \mathbb{R}^d$$

with integration weights $w_{i,\delta}$ at integration nodes $x_{i,\delta}$. This is a straightforward way to approximate f by a trial function of the form (3.4).

The error bound (7.4) now gets an additional term for the integration error. Near each $x \in \mathbb{R}^d$ there must be enough integration points to resolve the kernel at scale δ, and therefore the integration points must be closer than $\mathcal{O}(\delta)$. This approach will be called *stationary* below, and it needs more and more integration points for kernels of decreasing width.

But for reaching a prescribed accuracy, we can first choose a kernel scale δ such that (7.4) is sufficiently small. For this fixed δ we then perform a sufficiently good numerical integration to reproduce f sufficiently well by a finite linear combination of kernel translates. At this point, the smoothness of f and K determines the required density of integration points. This will be called a *nonstationary* setting below. This discussion will be resumed later, and it is crucial for the understanding of kernel approximations.

The *discretized convolution approach* leads to *quasi-interpolants* (Rabut 1989, Beatson and Light 1993, Buhmann *et al.* 1995, Maz'ya and Schmidt 2001, Ling 2005) of f which can be directly calculated from function values, without any linear system to solve. However, as is well known from the classical Bernstein or Schoenberg operators, there are better approximations using the same trial space. These will be dealt with later, but we note that quasi-interpolation works in quite a general fashion and is worth further investigation.

The theoretical consequence is that approximations from spaces spanned by translates of kernels result from an interaction between the scale of the kernel and the density of the translation points. This is a crucial issue for all kernel-based techniques, and it has consequences not only for the approximation, but also for its stability.

7.3. Fill distance and separation radius

In studying the approximation and stability properties of meshless methods, the following two geometric quantities are usually employed. Suppose we are confronted with a bounded set $\Omega \subseteq \mathbb{R}^d$ and a finite subset $X = \{x_1, \ldots, x_N\} \subseteq \Omega$ used for defining a trial space

$$K_X := \operatorname{span}\{K(\cdot, x_j) \ : \ x_j \in X\} \subset \mathcal{K}_0 \subset \mathcal{K} \tag{7.5}$$

in the terminology of Section 2. The *approximation power* of K_X is measured in terms of the *fill distance* of X in Ω, which is given by the radius of the largest data-site free ball in Ω, *i.e.*,

$$h_X := h_{X,\Omega} := \sup_{x \in \Omega} \min_{1 \le j \le N} \|x - x_j\|_2. \tag{7.6}$$

The second geometric quantity is the *separation radius* of X, which is half the distance between the two closest data sites, *i.e.*,

$$q_X := \tfrac{1}{2} \min_{j \neq k} \|x_j - x_k\|_2, \tag{7.7}$$

and does not depend on the domain. Obviously, the separation radius plays an important role in the stability analysis of the interpolation process, since a small q_X means that at least two points, and hence two rows in the system (3.5) are nearly the same. If the data in these two points are roughly the same or only differ by noise, it is reasonable to discard one of them. This is an instance of Guideline 3.17 and will be used by thinning algorithms within multiscale methods, as described in Section 10.

Finally, we will call a sequence of data sets $X = X_h$ *quasi-uniform* if there is a constant $c_q > 0$ independent of X such that

$$q_X \leq h_{X,\Omega} \leq c_q q_X. \tag{7.8}$$

The *mesh ratio* $\rho = \rho_{X,\Omega} := h_{X,\Omega}/q_X \geq 1$ provides a measure of how uniformly points in X are distributed in Ω. Remember that special results on convergence of univariate polynomial splines are valid only for cases with bounded mesh ratio; similar restrictions should be expected here as well.

7.4. Nonstationary versus stationary scales

There are two fundamentally different ways in which scales of kernel-based trial spaces are used in theory and practice. This often leads to misunderstandings of certain results, and therefore we have to be very explicit at this point.

In classical finite element and spline theory, the support of the nodal basis functions scales with the size of the mesh. For example, using classical hat functions to express a piecewise linear spline function over the node set $h\mathbb{Z}$ leads to a representation of the form

$$s_h(x) = \sum_{j\in\mathbb{Z}} \alpha_j B_1\big(\tfrac{x-jh}{h}\big) = \sum_{j\in\mathbb{Z}} \alpha_j B_1\big(\tfrac{x}{h} - j\big), \tag{7.9}$$

where B_1 is the standard hat function, which is zero outside $[0, 2]$ and is defined to be $B_1(x) = x$ for $0 \leq x \leq 1$ and $B_1(x) = 2 - x$ for $1 \leq x \leq 2$.

From (7.9) it follows that each of the basis functions $B_1(\frac{\cdot}{h} - j)$ has support $[jh, (j+2)h]$, *i.e.*, the support scales with the grid width. As a consequence, when setting up an interpolation system, each row in the interpolation matrix has the same number of nonzero entries (here actually only one); and this is independent of the current grid width. Hence, such a situation is usually referred to as a *stationary scheme.* Thus, for a stationary setting, the basis function scales linearly with the grid width.

In contrast to this, a *nonstationary scheme* keeps the basis function fixed for all fill distances h, *i.e.*, the approximant now takes the form

$$s_h(x) = \sum_{j\in\mathbb{Z}} \alpha_j B_1(x - jh), \tag{7.10}$$

resulting in a denser and denser interpolation matrix if h tends to zero.

Note that for univariate polynomial spline spaces these two settings generate the same trial space. But this is not true for general kernels. In any case of kernel usage, one should follow Guideline 3.15 and introduce a scaling parameter δ to form a *scaled* trial space $K_{\delta,X}$ as in (7.5). Then a *stationary* setting scales δ proportional to the fill distance $h_{X,\Omega}$ of (7.6), while the *nonstationary setting* uses a fixed δ and varies $h_{X,\Omega}$ only.

7.5. Stationary scales

A stationary setting arises with a discretization of the convolution approximation (7.1) if using integration points whose fill distance h is proportional to the kernel width δ. It is also the standard choice for finite element methods, including their generalized version (Babuška *et al.* 2003) and large classes of meshless methods with nodal bases (see Section 12).

The standard analysis tools for stationary situations are Strang–Fix conditions for the case of gridded data, while for general cases the Bramble–Hilbert lemma is applied, relying on reproduction of polynomials. These tools do not work in the nonstationary setting.

Stationary settings based on (fixed, but shifted and scaled) nodal functions with compact support will generate matrices with a *sparsity* which is independent of the scaling or fill distance. For finite element cases, the condition of the matrices grows like some negative power of h, but can be kept constant under certain conditions by modern preconditioning methods. But this does not work in general.

Guideline 7.2. For kernel methods, stationary settings have to be used with caution.

Without modification, the interpolants from Section 3 using stationary kernel-based trial spaces on regular data will not converge for $h \to 0$ for absolutely integrable kernels (Buhmann 1988, 1990), including the Gaussian and Wendland's compactly supported functions. But if kernels have no compact support, stationary kernel matrices will not be sparse, giving away one of the major advantages of stationary settings. There are certain methods to overcome this problem, and we shall deal with them in Section 8.

However, the practical situation is not as bad. Nobody can work for extremely small h anyway, such that convergence for $h \to 0$ is a purely theoretical issue. We summarize the experimental behaviour (Schaback 1997) as follows.

Guideline 7.3. The error of stationary interpolation by kernel methods decreases with $h \to 0$ to some small positive threshold value. This value can be made smaller by increasing the starting scale of the kernel, *i.e.*, by using a larger sparsity.

This effect is called *approximate approximation* (Maz'ya 1994, Lanzara, Maz'ya and Schmidt 2005) and deserves further study, including useful bounds of the threshold value. It is remarkable that it occurred first in the context of parabolic equations.

Practical work with kernels should follow Guideline 3.15 and adjust the kernel scale experimentally. Once it is fixed, the nonstationary setting applies, and this is how we argued in the case of discretized convolution above, if a prescribed accuracy is required. We summarize as follows.

Guideline 7.4. In meshless methods using positive definite kernels, approximation orders refer in general to a nonstationary setting. However, nonstationary schemes lead to ill-conditioned interpolation matrices. On the other hand, a fully stationary scheme generally provides no convergence but interpolation matrices with a condition number being independent of the fill distance.

Guideline 7.4 describes another general trade-off or uncertainty principle in meshless methods; see also Guideline 3.13. As a consequence, when working in practice with scaled versions of a single translation-invariant kernel, the scale factor needs special care. This brings us back to what we said about scaling in Sections 2 and 3, in particular Guideline 2.5.

However, from now on we shall focus on the nonstationary case.

7.6. Nonstationary interpolation

While in classical spline theory nonstationary approximants of the form (7.10) play no role at all, they are crucial in meshless methods for approximating and interpolating with positive definite kernels. Thus we now study approximation properties of interpolants of the form (3.4) with a *fixed* kernel but for various data sets X. To make the dependence on X and $f \in C(\Omega)$ explicit, we will use the notation

$$s_{f,X} = \sum_{j=1}^{N} \alpha_j K(\cdot, x_j),$$

where the coefficient vector is determined by the interpolation conditions $s_{f,X}(x_j) = f(x_j)$, $1 \le j \le N$ and the linear system (3.5) involving the kernel matrix $A_{K,X}$.

Early convergence results and error bounds were restricted to target functions $f \in \mathcal{K}$ from the native function space $\mathcal{K}$ of (2.7) associated to the employed kernel (Madych and Nelson 1988, 1990, 1992, Wu and Schaback 1993, Light and Wayne 1998). They are local pointwise estimates of the form

$$|f(x) - s_{f,X}(x)| \le CF(h)\|f\|_{\mathcal{K}}, \tag{7.11}$$

where F is a function depending on the kernel. For kernels of limited smoothness it is of the form $F(h) = h^{\beta/2}$, where β relates to the smoothness of K in the sense of Table 6.1. For infinitely smooth kernels such as Gaussians or (inverse) multiquadrics it has the form $F(h) = \exp(-c/h)$. A more detailed listing of kernels and their associated functions F can be found in the literature (Schaback 1995*b*, Wendland 2005*b*).

Guideline 7.5. If the kernel K and the interpolated function f are smooth enough, the obtainable approximation rate for nonstationary interpolation increases with the smoothness, and can be exponential in the case of analytic kernels and functions.

This supports Guideline 3.9 and is in sharpest-possible contrast to the stationary situation and finite element methods. It can also be observed when nonstationary trial spaces are used for solving partial differential equations.

Guideline 7.6. If the kernel K and the interpolated function f have different smoothness, the obtainable approximation rate for nonstationary interpolation depends on the smaller smoothness of the two.

Recent research has concentrated on the *escape scenario*, in which the smoothness of the kernel K exceeds the smoothness of f, *i.e.*, error estimates have to be established for target functions from outside the native Hilbert space. This is a realistic situation in applications, where a fixed kernel has to be chosen without knowledge of the smoothness of f. Surprisingly, these investigations have led far beyond kernel-based trial spaces.

To make this more precise, let us state two recent results (Narcowich, Ward and Wendland 2005*b*, 2004). As usual we let $W_p^k(\Omega)$ denote the Sobolev space of measurable functions having weak derivatives up to order k in $L_p(\Omega)$. Furthermore, we will employ *fractional* order Sobolev spaces $W_p^\tau(\Omega)$, which can, for example, be introduced using interpolation theory.

Theorem 7.7. Let k be a positive integer,

$$0 \le s < 1, \quad \tau = k + s, \quad 1 \le p < \infty, \quad 1 \le q \le \infty,$$

and let $m \in \mathbb{N}_0$ satisfy $k > m + d/p$ or, for $p = 1, k \ge m + d$. Let $X \subset \Omega$ be a discrete set with mesh norm $h_{X,\Omega}$ where Ω is a compact set with Lipschitz boundary which satisfies an interior cone condition. If $u \in W_p^\tau(\Omega)$ satisfies $u|_X = 0$, then

$$|u|_{W_q^m(\Omega)} \le C h_{X,\Omega}^{\tau - m - d(1/p-1/q)_+} |u|_{W_p^\tau(\Omega)},$$

where C is a constant independent of u and $h_{X,\Omega}$, and $(x)_+ = \max\{x, 0\}$.

Theorem 7.7 bounds lower Sobolev seminorms of functions in terms of a higher Sobolev seminorm, provided the functions have lots of zeros. It is entirely independent of any reconstruction method or trial space, and it can be successfully applied to any interpolation method that keeps a discretization-independent upper bound on a high Sobolev seminorm.

In fact, if $s_{f,X} \in W_p^\tau(\Omega)$ is an arbitrary function which interpolates $f \in W_p^\tau(\Omega)$ exactly in X, we have

$$|f - s_{f,X}|_{W_q^m(\Omega)} \le C h_{X,\Omega}^{\tau - m - d(1/p-1/q)_+} (|f|_{W_p^\tau(\Omega)} + |s_{f,X}|_{W_p^\tau(\Omega)}),$$

and if the interpolation manages to keep $|s_{f,X}|_{W_p^\tau(\Omega)}$ bounded independent of X, this is an error bound and an optimal order convergence result. This opens a new way to deal with all interpolation methods that are regularized properly.

For interpolation by kernels we can use Theorem 3.4 to provide the bound $|s_{f,X}|_{\mathcal{K}} \le |f|_{\mathcal{K}}$ if the kernel's native Hilbert space $\mathcal{K}$ is continuously embedded in Sobolev space $W_2^\tau(\Omega)$ and contains f. By embedding, we also have $|s_{f,X}|_{W_p^\tau(\Omega)} \le C|f|_{\mathcal{K}}$ and Theorem 7.7 yields error estimates of the form

$$\begin{aligned} |f - s_{f,X}|_{W_2^m(\Omega)} &\le C h_{X,\Omega}^{\tau-m} \|f\|_{\mathcal{K}}, \\ |f - s_{f,X}|_{W_\infty^m(\Omega)} &\le C h_{X,\Omega}^{\tau-m-d/2} \|f\|_{\mathcal{K}}. \end{aligned}$$

This still covers only the situation of target functions from the native Hilbert space, but it illustrates the regularization effect provided by Theorem 3.4 and described in Guideline 3.12.

The next result is concerned with the situation that the kernel's native space is norm-equivalent to a smooth Sobolev space $W_2^\tau(\Omega)$ while the target function comes from a rougher Sobolev space $W_2^\beta(\Omega)$. It employs the mesh ratio $\rho_{X,\Omega} = h_{X,\Omega}/q_X$.

Theorem 7.8. If $\tau \ge \beta$, $\beta = k + s$ with $0 < s \le 1$ and $k > d/2$, and if $f \in W_2^\beta(\Omega)$, then

$$\|f - s_{f,X}\|_{W_2^\mu(\Omega)} \le C h_{X,\Omega}^{\beta-\mu} \rho_{X,\Omega}^{\tau-\mu} \|f\|_{W_2^\beta(\Omega)}, \quad 0 \le \mu \le \beta.$$

In particular, if X is quasi-uniform, this yields

$$\|f - s_{f,X}\|_{W_2^\mu(\Omega)} \le C h_{X,\Omega}^{\beta-\mu} \|f\|_{W_2^\beta(\Omega)}, \quad 0 \le \mu \le \beta. \tag{7.12}$$

Note that these error bounds are of optimal order. Furthermore, since they can be applied locally, they automatically require fewer data or boost the approximation order at places where the function is smooth.

Guideline 7.9. Nonstationary kernel approximations based on sufficiently smooth kernels have both h- and p-adaptivity.

7.7. Condition

But this superb approximation behaviour comes at the price of ill-conditioned matrices, if no precautions like preconditioning (Faul and Powell 1999, Ling and Kansa 2004, Brown *et al.* 2005, Ling and Kansa 2005) are taken. This is due to the fact that rows and columns of the kernel matrix $A_{K,X}$ with entries $K(x_i, x_j)$ relating to two close points x_i and x_j will be very similar. Thus the condition will increase when the separation radius q_X of (7.7) decreases, even if the fill distance $h_{X,\Omega}$ is kept constant, *i.e.*, when adding data points close to existing ones.

A thorough analysis shows that the condition number of the kernel matrix $A_{K,X} = (K(x_i, x_j))$ depends mainly on the smallest eigenvalue of $A_{K,X}$, while the largest usually does not increase more rapidly than the number N of data points. For the smallest eigenvalue it is known (Narcowich and Ward 1991, Ball 1992, Ball, Sivakumar and Ward 1992, Binev and Jetter 1992, Narcowich and Ward 1992, 1994*b*, Schaback 1995*b*, Wendland 2005*b*) that it can be bounded from below by

$$\lambda_{\min}(A_{K,X}) \geq cG(q_X).$$

Unfortunately, in many cases this inequality is sharp and the function G is related to the function F arising in (7.11) by $G(q) = \Theta(F(q^2))$ for $q \to 0$ (Schaback 1995*b*). This is the theoretical background of Guideline 3.13 relating error and condition.

7.8. Approximation via relaxed interpolation

The above discussion suggests the following heuristic.

Guideline 7.10. The best approximation error with the most stable system is achieved by using quasi-uniform data sets.

Sorting out nearly coalescing points by thinning (Floater and Iske 1998) and going over to suitable subproblems by adaptive methods (Schaback and Wendland 2000*a*, Bozzini, Lenarduzzi and Schaback 2002, Hon, Schaback and Zhou 2003, Ling and Schaback 2004, de Marchi, Schaback and Wendland 2005) are useful to ensure quasi-uniformity. However, these methods are dangerous in the presence of noise.

One possible remedy to both problems, coalescing points and noisy data, is to relax the interpolation condition and to solve instead the following smoothing problem:

$$\min\left\{\sum_{j=1}^{N}[f(x_j) - s(x_j)]^2 + \lambda\|s\|_{\mathcal{K}}^2, : s \in \mathcal{K}\right\}, \tag{7.13}$$

where $\lambda > 0$ is a certain smoothing parameter balancing the resulting approximant between interpolation and approximation. This problem occurred in a probabilistic setting in Section 4. It is also intensively studied in the context of kernel learning (Cristianini and Shawe-Taylor 2000, Cucker and Smale 2001, Schölkopf and Smola 2002, Shawe-Taylor and Cristianini 2004) and in the theory of regularization networks (Evgeniou, Pontil and Poggio 2000).

A standard result of central importance for all noisy recovery problems and including learning theory is the following generalization of Theorem 3.4.

Theorem 7.11. Suppose K is the reproducing kernel of the Hilbert space $\mathcal{K}$. Then the solution to (7.13) is given by a function of the form (3.4), where the coefficient vector $\alpha = \{\alpha_j\}$ now can be calculated by solving the linear system

$$(A_{K,X} + \lambda I)\alpha = f|X.$$

Guideline 7.12. Relaxed interpolation along the lines of (7.13) is computationally equivalent to recovery from noisy observations. The relaxation parameter λ is connected to the noise variance σ by $\lambda = \sigma^2$.

Theorem 7.11 shows that the ill-conditioning problem is simply addressed by moving the eigenvalues of the interpolation matrix away from zero by an offset given by the smoothing parameter $\lambda > 0$.

However, this immediately introduces the problem of how to choose the smoothing parameter. There have been thorough investigations mainly motivated by probabilistic approaches along the lines of Section 4 (Reinsch 1967, 1971, Wahba 1975, Craven and Wahba 1979, Ragozin 1983, Cox 1984, Wahba 1990, Wei, Hon and Wang 2005).

Instead of going into details here, we follow a recent deterministic approach (Wendland and Rieger 2005) which is based upon the following simple observation. The solution s_λ of (7.13) allows the following bounds:

$$\begin{aligned} |f(x_j) - s_\lambda(x_j)| &\le \sqrt{\lambda}\|f\|_{\mathcal{K}} \quad \text{for all } 1 \le j \le N, \\ \|s_\lambda\|_{\mathcal{K}} &\le \|f\|_{\mathcal{K}}. \end{aligned}$$

Both can easily be verified, since $f \in \mathcal{K}$ is feasible in (7.13). Hence, if λ is considered to be small, the error function $u = f - s_\lambda$ is approximately zero on X and its $\mathcal{K}$-norm can be bounded by twice the $\mathcal{K}$-norm of f.

Theorem 7.13. Assume that all assumptions of Theorem 7.7 hold, except for $u|X = 0$. Then the following generalized estimate holds:

$$|u|_{W_q^m(\Omega)} \le C\big(h_{X,\Omega}^{\tau-m-d(1/p-1/q)_+}|u|_{W_p^\tau(\Omega)} + h_{X,\Omega}^{-m}\|u|X\|_\infty\big). \tag{7.14}$$

This is a generalization of a Poincaré–Friedrichs inequality, and it turns out to be very useful for the analysis of unsymmetric kernel-based methods for solving partial differential equations (Schaback 2005*a*). Under the assumptions of Theorem 7.8 this yields the estimate

$$\|f - s_\lambda\|_{L_\infty(\Omega)} \le C\big(h_{X,\Omega}^{\tau-d/2} + \sqrt{\lambda}\big)\|f\|_{\mathcal{K}}.$$

for our smoothing problem. Keeping in mind that in this particular situation $F(h) = h^{\tau-d/2}$ and $G(q) = q^{2\tau-d}$, we have the following guideline.

Guideline 7.14. If the smoothing parameter $\lambda > 0$ is chosen as $\lambda = Ch^{2\tau-d}$, the relaxed technique still has an optimal order of approximation, while the smallest eigenvalue now behaves as in the case of quasi-uniformity.

In contrast to adaptive methods working on smaller subproblems, this relaxed approximation will still have a full set of coefficients in (3.4). It is a challenging open problem to prove deterministic results concerning the complexity reduction obtainable by a more general relaxation like (3.9).

7.9. Moving least squares

While our analysis in the previous subsections dealt with nonstationary approximation schemes based on kernel methods, we will now discuss a particular *stationary* scheme. Approximation by moving least squares has a long history (Shepard 1968, McLain 1974, 1976, Lancaster and Salkauskas 1981, Farwig 1986, 1987, 1991). It has become popular again in approximation theory (Levin 1999, Wendland 2001), in computer graphics (Mederos, Velho and de Figueiredo 2003, Fleishman, Cohen-Or and Silva 2005), and in meshless methods for solving partial differential equations (Belytschko *et al.* 1996*b*).

The idea of moving least squares approximation is to solve for every point x a locally weighted least squares problem, where a kernel is used as a weight function. This appears to be quite expensive at first sight, but actually it is a very efficient method, because it can come at constant cost per evaluation, independent of the number and complexity of the data. Moving least squares also arise in meshless methods, where they are used for a ‘nodal basis’ to generate data in nearby locations, *e.g.*, for performing the integrations to calculate entries of a stiffness matrix. Moreover, in many applications we are interested in only a few evaluations. For such cases, moving least squares techniques are even more attractive, because it is not necessary to set up and solve a large system.

The influence of the data points is governed by a weight function $w : \Omega \times \Omega \to \mathbb{R}$, which becomes smaller the farther its arguments are away from each other. Ideally, w vanishes for arguments $x, y \in \Omega$ with $\|x - y\|_2$ greater than a certain threshold. Such behaviour can be modelled by using a translation-invariant nonnegative kernel of compact support, with no need for positive definiteness. As in any other kernel-based method, Guideline 2.5 makes *scaling* a serious issue, and Guideline 3.15 implies that w should be of the form $w(x, y) = \Phi_\delta(x - y)$ with a controllable scaled version $\Phi_\delta = \Phi(\cdot/\delta)$ of a compactly supported kernel $\Phi : \mathbb{R}^d \to \mathbb{R}$.

Definition 7.15. For $x \in \Omega$ the value $s_{f,X}(x)$ of the moving least squares approximant is given by $s_{f,X}(x) = p^*(x)$ where p^* is the solution of

$$\min\left\{\sum_{i=1}^{N}(f(x_i) - p(x_i))^2 w(x, x_i) \;:\; p \in \pi_m(\mathbb{R}^d)\right\}. \tag{7.15}$$

Here, $\pi_m(\mathbb{R}^d)$ denotes the space of all d-variate polynomials of degree at most m. But it is not at all necessary to restrict oneself to polynomials.

It is, for example, even possible to incorporate singular functions into the finite-dimensional function space. This is a common trick for applications in meshless methods dealing with shocks and cracks in mechanics (Belytschko, Krongauz, Fleming, Organ and Liu 1996*a*).

The minimization problem (7.15) can be seen as a discretized version of the continuous problem

$$\min\biggl\{\int_{\mathbb{R}^d} |f(y)-p(y)|^2 w(x,y)\,\mathrm{d}y : p \in \pi_m(\mathbb{R}^d)\biggr\},$$

where the integral is supposed to be restricted by the support of the weight function to a region around the point x.

The simplest case of (7.15) is given by choosing only constant polynomials, *i.e.*, $m=0$. In this situation, the solution of (7.15) can easily be computed to the explicit form

$$s_{f,X}(x) = \sum_{j=1}^{N} f(x_j)\tfrac{w(x,x_j)}{\sum_{k=1}^{N} w(x,x_k)}, \tag{7.16}$$

which is also called *Shepard approximant* (Shepard 1968). From the explicit form (7.16), one can already read off some specific properties, which also hold more generally for moving least squares. First of all, since the weight function $w(x,y)$ is supposed to be nonnegative, so are the 'basis' functions

$$u_j(x) = \tfrac{w(x,x_j)}{\sum_{k=1}^{N} w(x,x_k)}, \quad 1 \le j \le N,$$

which also occur under the name 'shape functions' or 'particle functions' in meshless methods (see Section 12). Moreover, these functions form a *partition of unity*, *i.e.*, they satisfy

$$\sum_{j=1}^{N} u_j(x) = 1. \tag{7.17}$$

Note that partitions of unity arise again in Section 8, and they play an important role in computer-aided design because of their invariance under affine transformations.

Nonnegativity and partition of unity already guarantee linear convergence, if the weight functions are of the form $w(x,y)=\Phi((x-y)/h)$, since we have for all $p \in \pi_0(\mathbb{R}^d)$

$$\begin{aligned} |f(x)-s_{f,X}(x)| &\le |f(x)-p(x)| + |p(x)-s_{f,X}(x)| \\ &\le |f(x)-p(x)| + \sum_{j=1}^{N} u_j(x)|p(x)-f(x_j)| \\ &\le 2\|f-p\|_{L_\infty(B(x,h))} \end{aligned}$$

and the last term can be bounded by $C_f h$ if p is the local Taylor polynomial to f of degree zero.

To derive a similar result for the general moving least squares approximation scheme, it is important to rewrite the approximant in form of a quasi-interpolant

$$s_{f,X}(x) = \sum_{j=1}^{N} u_j^m(x) f(x_j),$$

as in Theorem 3.6. This is always possible under mild assumptions on the data sites. Though the basis functions $u_j^m(x)$ are in general not nonnegative, they satisfy a constrained minimization problem, which leads to a uniform bound of the ℓ_1-norm of $\{u_j^m(x)\}_{j=1}^N$. From this, convergence orders can be derived. The following result (Wendland 2001) summarizes this discussion.

Theorem 7.16. Suppose the data set $X \subseteq \Omega$ is quasi-uniform and $\pi_m(\mathbb{R}^d)$-unisolvent. If the support radius δ of the compactly supported, nonnegative weight function $w(x,y) = \Phi((x-y)/\delta)$ is chosen proportional to the fill distance $h_{X,\Omega}$ and if $f \in C^{m+1}(\mathbb{R}^d)$ is the target function, then the error can be bounded by

$$\|f - s_{f,X}\|_{L_\infty(\Omega)} \le C_f h_{X,\Omega}^{m+1}.$$

It is remarkable that this result actually is local, *i.e.*, in regions where the target function is less smooth, the associated approximation order is automatically achieved. As in Guideline 7.9 we have the following.

Guideline 7.17. Moving least-squares methods have both h- and p-adaptivity, if the order m of the local polynomial space is large enough and if sufficiently many local data points are included.

Moreover, the assumption on the quasi-uniformity of the data set can be dropped if the support radius is continuously adapted to the local fill distance.

Finally, if for a point $x \in \Omega$ the positions of a bounded number of surrounding data sites in the ball of radius $\delta = Ch_{X,\Omega}$ are known, the minimization problem can be solved and hence the moving least squares approximation can be computed in constant time. Locating the relevant data sites can be done by employing an 'intelligent' data structure in at most $\mathcal{O}(\log N)$ time, if an additional $\mathcal{O}(N \log N)$ time is allowed to build the data structure. This, of course, is only relevant if a substantial number of evaluations is necessary. For only a few evaluations, all relevant data sites can be found by brute force methods in linear time.

8. Large and ill-conditioned kernel systems

Section 7 indicated that approximation by nonstationary scales of kernel-based trial spaces may lead to large, non-sparse systems which are often highly ill-conditioned. This will become important for applications in Section 10. Hence, it is now time to discuss efficient methods for solving large and dense systems arising from kernel approximations or interpolations. Note that these systems are qualitatively different from those arising in finite element methods (see Guidelines 3.17 and 3.18), and thus they call for different numerical techniques.

There are five major approaches in this area:

- multipole expansions, often coupled with
- domain decomposition methods,
- partition of unity methods,
- multilevel techniques using compactly supported kernels,
- preconditioning.

Each of these methods has its strengths and drawbacks and it depends on the users to decide which one suits their application best.

8.1. Multipole expansions

We start with the discussion of multipole expansions. They are, in the first place, only a tool to approximately evaluate sums of the form

$$s(x) = \sum_{j=1}^{N} \alpha_j K(x, x_j) \tag{8.1}$$

from (3.4) in a fast way. As a matter of fact, they have been developed in the context of the N-body problem, which appears in various scientific fields (Barnes and Hut 1986, Appel 1985, Greengard and Rokhlin 1987).

Large systems of the form (3.5) cannot be solved by any direct method. Instead, iterative methods have to be employed. No matter which iterative method is used, the main operation is a matrix by vector multiplication, which is nothing but the evaluation of N sums of the form (8.1).

Hence, not only for a fast evaluation of the interpolant or approximant but also for solving the linear equations (3.5) it is crucial to know how to calculate the above sums efficiently.

To derive a sufficiently fast evaluation of (8.1), for every evaluation point x the sum is split into the form

$$s(x) = \sum_{j \in I_1} \alpha_j K(x, x_j) + \sum_{j \in I_2} \alpha_j K(x, x_j), \tag{8.2}$$

where I_1 contains the indices of those points x_j that are close to x, while

I_2 contains the indices of those points x_j that are far away from x. Both sums can now be replaced by approximations to them. In the first case, since $\|x - x_j\|_2$ is small for $x_j \in I_1$, the associated sum can, for example, be approximated by a Taylor polynomial. This is sometimes called a *near-field expansion*. More important is a proper approximation to the second sum, which is done by a *unipole* or *far-field* expansion.

The main idea of such an expansion is based upon a kernel expansion of the form (5.1). Incorporating the weights w_i into the function φ_i and also allowing different functions for the two arguments, this can more generally be written as

$$K(x,t) = \sum_{i=1}^{\infty} \varphi_i(x)\psi_i(t), \tag{8.3}$$

and we usually refer to t in $\Phi(x,t)$ as a *source point*, while x is called an *evaluation point*.

Now suppose that the source points x_j, $j \in I_2$, are located in a local *panel* with centre t_0, which is sufficiently far away from the evaluation point, *i.e.*, panel and evaluation points are *well separated*. Suppose further, (8.3) can be split into

$$K(x,t) = \sum_{k=1}^{p} \phi_k(x)\psi_k(t) + R_p(x,t) \tag{8.4}$$

with a remainder R_p that tends to zero for $\|x - t_0\|_2 \to \infty$ or for $p \to \infty$ if $\|x - t_0\|_2$ is sufficiently large. Then, (8.4) allows us to evaluate the second sum s_2 in (8.2) by

$$\begin{aligned} s_2(x) &:= \sum_{j \in I_2} \alpha_j K(x, x_j) \\ &= \sum_{j \in I_2} \alpha_j \sum_{k=1}^{p} \phi_k(x)\psi_k(x_j) + \sum_{j \in I_2} \alpha_j R(x, x_j) \\ &= \sum_{k=1}^{p} \phi_k(x) \sum_{j \in I_2} \alpha_j \psi_k(x_j) + \sum_{j \in I_2} \alpha_j R(x, x_j) \\ &=: \sum_{k=1}^{p} \beta_k \phi_k(x) + \sum_{j \in I_2} \alpha_j R(x, x_j). \end{aligned}$$

Hence, if we use the approximation $\widetilde{s_2}(x) = \sum_{k=1}^{p} \beta_k \phi_k(x)$ we have an error bound

$$|s_2(x) - \widetilde{s_2}(x)| \le \|\alpha\|_1 \max_{j \in I_2} |R(x, x_j)|,$$

which is small if x is sufficiently far away from the sources x_j, $j \in I_2$.

Moreover, each coefficient β_k can be computed in advance in linear time. Thus, if p is much smaller than N, we can consider it as constant and we need only constant time for each evaluation of $\widetilde{s_2}$.

So far, we have developed an efficient method for evaluating a sum of the form (8.1) for one evaluation point or, more generally, for evaluation points from the same panel, which is well separated from the panel containing the source points. To derive a fast summation formula for arbitrary evaluation points $x \in \Omega$, the idea has to be refined. To this end, the underlying region of interest Ω is subdivided into cells or panels. To each panel a far field and a near field expansion is assigned. For evaluation, all panels are visited and, depending on whether or not the panel is well separated from the panel which contains the evaluation point, the near field or far field expansion is used.

The decomposition of Ω into panels can be done either uniformly or adaptively, dependent on the data. A uniform decomposition makes a near field expansion indispensable since the cardinality of I_1 cannot be controlled. However, its simple structure makes it easy to implement and hence it has been and is still often used. An adaptive decomposition is usually based on a tree-like data structure where the panels are derived by recursive subdivision of space. More details can be found in the literature (Greengard and Strain 1991, Beatson and Newsam 1992, Beatson, Goodsell and Powell 1996, Beatson and Greengard 1997, Beatson and Light 1997, Beatson and Newsam 1998, Roussos 1999, Beatson and Chacko 2000, Beatson, Cherrie and Ragozin 2000*a*, 2001).

In any case, since we now have to implement a unipole expansion for every panel, the resulting technique is called *multipole expansion*.

Unfortunately, the multipole expansion has to be precomputed for each kernel separately. However, for translation-invariant kernels $K(x, y) = K(x - y)$, it suffices to know the far field expansion around zero. Because this gives the far field expansion around any t_0 simply by

$$\begin{aligned} K(x-t) &= K((x-t_0)-(t-t_0)) \\ &= \sum_{k=1}^{p} \phi_k(x-t_0)\psi_k(t-t_0) + R(x-t_0, t-t_0). \end{aligned}$$

The far field expansion around zero can often be calculated using Laurent expansions of the translation-invariant kernel. Details can be found in the literature cited above.

8.2. *Domain decomposition*

Having a fast evaluation procedure for functions of the form (8.1) at hand, different iterative methods for solving the linear system (3.5) can be applied. However, the reader should be aware of the fact that the far field expansion may now lead to a nonsymmetric situation (Beatson, Cherrie and Mouat 1999).

Here, we want to describe a domain decomposition method (Beatson, Light and Billings 2000*b*), which can be extended to generalized interpolation problems (Wendland 2004). Domain decomposition is a standard technique in finite elements, involving interface conditions and related to Schwarz iteration. In kernel techniques, it is much simpler, has a fundamentally different flavour and already quite some history in the context of meshless methods for partial differential equations (Dubal 1994, Hon and Wu 2000*a*, Zhou, Hon and Li 2003, Ingber, Chen and Tanski 2004, Li and Hon 2004, Ling and Kansa 2004). However, the name is rather misleading here, since the domain or the analytic problem are not decomposed, but rather the approximation or the trial space. The technique itself is an iterative projection method.

To decompose the trial space it suffices to decompose the set of centres X, or generally the set of data functionals in the sense of (3.8). To be more precise, let us decompose X into subsets $X_1, \dots, X_k$. These subsets need not be disjoint but their union must be X. Then the algorithm starts to interpolate on the first set X_1, forms the residual, interpolates this on X_2 and so on. After k steps one cycle of the algorithm is complete and it starts over again. A more formal description is

```
(1) Set $f_0 := f$, $s_0 := 0$.
(2) For $n = 0, 1, 2, \dots$
        For $r = 1, \dots, k$
            $f_{nk+r} := f_{nk+r-1} - s_{f_{nk+r-1},X_r}$
            $s_{nk+r} := s_{nk+r-1} + s_{f_{nk+r-1}X_r}$,
        If $\|f_{(n+1)k}\|_{L_\infty(X)} < \epsilon$ stop.
```

This algorithm approximates the interpolant $s_{f,X} = f^*$ from (3.4) up to the specified accuracy. The convergence result is based upon the fact that the interpolant $s_{f,X}$ is also the *best approximant* to f from the subspace K_X of (7.5) in the native Hilbert space norm. This optimality is another instance of Guideline 3.1 which we suppressed in Section 3 for brevity.

Convergence is achieved under very mild assumptions on the decomposition. The data sets X_j have to be *weakly disjoint* meaning that $X_j \neq Y_j$ and $Y_{j+1} \neq Y_j$ for each $1 \leq j \leq k-1$, where $Y_j = \cup_{i=j}^k X_i$, $1 \leq j \leq k$. This is, for example, satisfied, if each X_j contains at least one data site, which is not contained in any other X_i.

Theorem 8.1. Let $f \in \mathcal{K}$ be given. Suppose $X_1, \dots, X_k$ are weakly distinct subsets of $\Omega \subseteq \mathbb{R}^d$. Set $Y_j = \cup_{i=j}^k X_i$, $1 \leq j \leq k$. Denote with $s^{(j)}$

the approximant after j completed cycles. Then there exists a constant $c \in (0,1)$ so that

$$\|f^* - s^{(n)}\|_{\mathcal{K}} \leq c^n \|f\|_{\mathcal{K}}.$$

For a proof of this theorem and for a more thorough discussion on how the subsets X_j have to be chosen we refer the reader to the literature (Beatson *et al.* 2000*b*, Wendland 2005*b*).

For an efficient implementation within multipole codes we need not only the far field or multipole expansion of the kernel. Since the coefficients of the sum (8.1) are now changing with every iteration, we also need intelligent update formulas. Finally, the decomposition of X into $X_1, \dots, X_k$ has to be done in such a way that the local interpolants and the (global) residuals can be computed efficiently.

Theorem 8.1 suggests a hidden connection to preconditioning, if the local problems are solved by approximate inverses of the local submatrices. But this is an open research question.

8.3. Partitions of unity

Any iterative method for solving the system (3.5) leads to a full $\mathcal{O}(N)$-term solution of the form (3.4). Unless the inverse of the kernel matrix is sparse, every data site x_k has some influence on every evaluation point x, even if compactly supported kernels are used. To improve *locality* in the sense of letting only nearby data locations x_j influence the solution at x, multipole methods are a possible choice. Moving least squares approximants have this property by definition, but they need recalculation at each new evaluation point, because they calculate values, not functions. *Partitions of unity* are a compromise, because they allow us to patch local approximating functions together into a global approximating function, allowing a cheap local function evaluation.

While the 'domain decomposition' methods above decompose the data set rather than the domain, we now actually decompose the domain $\Omega \subseteq \cup_{j=1}^{M} \Omega_j$ in an overlapping manner into simple small subdomains Ω_j which may, for instance, be Euclidean balls. Associated to this covering $\{\Omega_j\}$ we choose a partition of unity, *i.e.*, a family of weight functions $w_j : \Omega_j \to \mathbb{R}$, which are nonnegative, supported in Ω_j, and satisfy

$$\sum_{j=1}^{M} w_j(x) = 1, \quad x \in \Omega.$$

These weight functions are conveniently chosen as translates of *kernels* which are smooth and compactly supported, but not necessarily positive definite. If balls are used, and if the problem is isotropic, the kernels should be compactly supported *radial basis functions.*

Finally, we associate to each cell Ω_j an approximation space V_j and an approximation process which maps a function $f : \Omega_j \to \mathbb{R}$ to an approximation $s_j : \Omega_j \to \mathbb{R}$. This approximation process can, for example, be given by local interpolants using only the data sites $X_j = X \cap \Omega_j$. However, the whole procedure works for arbitrary approximation processes, *e.g.*, for approximations by augmented finite elements bases, thus leading to the *generalized finite element method* (Melenk and Babuška 1996, Babuška and Melenk 1997, Babuška, Banerjee and Osborn 2002, Babuška *et al.* 2003)

In the end, the global approximant is formed from the local approximants by weighting:

$$s(x) = \sum_{j=1}^{M} w_j(x) s_j(x), \quad x \in \Omega.$$

From the partition of unity property, we can immediately see that

$$\begin{aligned} |f(x) - s(x)| &= \left| \sum_{j=1}^{M} [f(x) - s_j(x)] w_j(x) \right| \\ &\le \sum_{j=1}^{M} |f(x) - s_j(x)| w_j(x) \\ &\le \max_{1 \le j \le M} \|f - s_j\|_{L_\infty(\Omega_j)} \end{aligned}$$

implies the following rule.

Guideline 8.2. The partition of unity approximant is at least as good as its worst local approximant.

More sophisticated error estimates can be found in the literature (Babuška and Melenk 1997, Wendland 2005*b*), also including bounds on the derivatives (simultaneous approximation). In the latter case, additional assumptions on the partitions and the weight functions have to be made. However, for an efficient implementation of the partition of unity method, these are automatically satisfied in general.

To control the complexity of *evaluating* the partition of unity approximant, the cells must not overlap too much, *i.e.*, every $x \in \Omega$ has to be contained in only a small number of cells and these cells must be easily determinable. Moreover, each local approximant has to be evaluated efficiently. Keeping Guideline 8.2 in mind, this often goes hand in hand with the fact that the regions are truly local, meaning that their diameter is of the size of the fill distance or the separation distance. For example, if the local approximation process employs polynomials, a diameter $\mathcal{O}(h_{X,\Omega})$ of local domains guarantees good approximation properties of the local approximants by a Taylor polynomial argument. If interpolation by kernels is employed, it is more important that the number of centres in each cell can be considered constant when compared to the global number of centres. In each case we have to assume that the number of cells is roughly proportional to the number of data sites. In this situation, all local interpolants can be computed in *linear* time provided that the local centres are known. Hence, everything depends upon a good data structure for both the centres and the cells, which can be provided by tree-like constructions again. We summarize as follows.

Guideline 8.3. Localization strategies within kernel methods should try to use a fixed or at least globally bounded number of data in each local domain. This applies to panels in multipole expansions, to domain decomposition methods, to partitions of unity, to preconditioning by local cardinal bases, and to all stationary methods.

The proper choice of scalings of kernels or influence regions is a major research area in theory, while proper programming and experimentation gives good practical results. Note that partitions of unity provide a localization strategy which helps with the scaling dilemma and mimics a stationary situation.

Finally, the easiest way to construct the partition of unity weight functions w_j is by employing moving least-squares in its simplest form, namely Shepard approximants (see Section 7).

8.4. Multilevel and compactly supported kernels

We now turn to a method tailored in particular to compactly supported kernels. We know from Section 7 that interpolation in the *stationary* setting will not lead to convergence. Moreover, to guarantee solvability, the same support radius for all basis functions has to be used. In a certain way, this contradicts a well-known rule from signal analysis as follows.

Guideline 8.4. Resolve coarse features by using a large support radius, and finer features with a smaller support radius.

To obey Guideline 8.4, the following multilevel scheme is useful. We first split our set X into a nested sequence

$$X_1 \subseteq X_2 \subseteq \cdots \subseteq X_k = X. \tag{8.5}$$

If X is quasi-uniform, meaning that the separation radius q_X of (7.7) has size comparable to the fill distance $h_{X,\Omega}$ of (7.6), then the subsets X_j should also be quasi-uniform. Moreover, they should satisfy $q_{X_{j+1}} \approx cq_{X_j}$ and $h_{X_{j+1},\Omega} \approx ch_{X_j,\Omega}$ with a fixed constant c.

Now the *multilevel method* (Floater and Iske 1996, Schaback 1997) is simply one cycle of the domain decomposition method. But this time we use compactly supported basis functions with a different support radius at each level. We could even use different basis functions at different levels. Hence, a general formulation proceeds as follows. For every $1 \le j \le k$ we choose a kernel K_j and form the interpolant

$$s_{f,X_j,K_j} = \sum_{x_j \in X_j} c_{x_j}(f) K_j(\cdot, x_j)$$

by using a kernel K_j on level j. We have in mind to take $K_j(x,y)$ as $K((x-y)/\delta_j)$ with a compactly supported basis function K and a scaling parameter δ_j proportional to $h_{X_j,\Omega}$. The idea behind this algorithm is that one starts with a very thin, widely spread set of points and uses a smooth basis function to recover the global behaviour of the function f. In the next level a finer set of points is used and a less smooth function possibly with a smaller support is employed to resolve more details, and so on.

As we said before, the algorithm performs one cycle of the domain decomposition algorithm:

```
set $f_0 = f$ and $s_0 = 0$.
for $1 \le j \le k$
    $s_j = s_{j-1} + s_{f_{j-1},X_j,K_j}$
    $f_j = f_{j-1} - s_{f_{j-1},X_j,K_j}$
```

The method shows linear convergence between levels (Schaback 1997), but a thorough theoretical analysis is a hard research problem with only partial results known (Narcowich, Schaback and Ward 1999, Hales and Levesley 2002).

8.5. Preconditioning

The localization techniques used above can be modified to enable specific preconditioning methods. Any good preconditioning technique must somehow implement an approximate inverse to the linear system to be solved. This can be done classically by partial *LU* or Cholesky factorization, but it can also be done by approximately inverting the matrices of the subproblems introduced by localization. This approximate inversion of local kernel matrices is a transition from the basis $K(\cdot, x_j)$ to local cardinal or Lagrangian functions, as (3.7) and (3.5) show.

Such methods are around for a while (Faul and Powell 1999, Mouat 2001, Schaback and Wendland 2000*b*) and have also been demonstrated to be quite effective within meshless kernel-based methods for solving partial differential equations (Ling and Kansa 2004, Brown *et al.* 2005, Ling and Kansa 2005). We have to leave details to the cited literature, but here is again a promising research field. In particular, considering the limit for wide-scaled analytic kernels reveals unexpected connections to polynomial interpolation (Schaback 2005*a*) and allows us to handle cases beyond all condition limits (Driscoll and Fornberg 2002, Larsson and Fornberg 2005).

9. Kernels on spheres

Expansions of the form (5.1) play a less important role for kernels defined on $\mathbb{R}^d$. There, continuous Fourier or Laplace transform techniques dominate the theory of characterizing and analysing such kernels.

The situation changes, if kernels on tori and spheres, or more generally, on compact (homogenous) Riemannian manifolds (Narcowich 1995, Dyn *et al.* 1999) are considered. There, expansions of the form (5.1) are natural. Also, the summation of feature functions in learning theory, as described in Section 5 and leading to Mercer kernels, is a standard application area for kernels defined by summation of products.

As a placeholder for more general situations, we will shortly outline the theory of approximation by kernels on the sphere

$$S^{d-1} := \{x \in \mathbb{R}^d : \|x\|_2 = 1\} \subseteq \mathbb{R}^d.$$

However, since there are some nice overview articles and books on approximation on the sphere including results on positive definite kernels (Freeden, Schreiner and Franke 1997, Freeden, Gervens and Schreiner 1998, Fasshauer and Schumaker 1998), and since this topic has also been covered in surveys and books on radial basis functions (Buhmann 2000, Wendland 2005*b*), we will restrict ourselves only to a basic introduction and some very recent results.

9.1. Spherical harmonics

On the sphere, the basis functions φ_i in (5.1) are given by *spherical harmonics* (Müller 1966). Here, we use the fact that a spherical harmonic is the restriction of a homogenous harmonic polynomial to the sphere. We will denote a basis for the set of all homogenous harmonic polynomials of degree ℓ by

$$\{Y_{\ell,k} : 1 \le k \le N(d,\ell)\},$$

where $N(d,\ell)$ denotes the dimension of this space. Moreover, the space of polynomials of degree at most L, restricted to the sphere $\pi_L(S^{d-1}) = \pi_L(\mathbb{R}^d)|S^{d-1}$ possesses the orthonormal basis

$$\{Y_{\ell,k} : 1 \le k \le N(d,\ell), 0 \le \ell \le L\}$$

and any $L_2(S^{d-1})$ function f can be expanded into a Fourier series

$$f = \sum_{\ell=0}^{\infty} \sum_{k=1}^{N(d,\ell)} \widehat{f}_{\ell,k} Y_{\ell,k} \quad \text{with } \widehat{f}_{\ell,k} = (f, Y_{\ell,k})_{L_2(S^{d-1})},$$

where $(\cdot,\cdot)_{L_2(S^{d-1})}$ is the usual $L_2(S^{d-1})$ inner product.

To understand and investigate *zonal* basis functions, which are the analogue of radial basis functions on the sphere (see Section 2), we need the well-known addition theorem for spherical harmonics. Between the spherical harmonics of order ℓ and the generalized Legendre polynomial $P_\ell(d;\cdot)$ of degree ℓ there exists the relation

$$\sum_{k=1}^{N(d,\ell)} Y_{\ell,k}(x)Y_{\ell,k}(y) = \tfrac{N(d,\ell)}{\omega_{d-1}} P_\ell(d; x^T y), \quad x, y \in S^{d-1}. \tag{9.1}$$

Here ω_{d-1} denotes the surface area of the sphere in $\mathbb{R}^d$.

9.2. Positive definite functions on spheres

After introducing spherical harmonics, we can write down the analogue of the kernel expansion (5.1) as

$$K(x,y) = \sum_{\ell=0}^{\infty} \sum_{k=1}^{N(d,\ell)} a_{\ell,k} Y_{\ell,k}(x)Y_{\ell,k}(y), \quad x, y \in S^{d-1}. \tag{9.2}$$

Such a kernel is obviously positive definite if all coefficients $a_{\ell,k}$ are positive, following Guideline 5.4 concerning positive transforms. Here and in the rest of this section we will assume that the coefficients decay sufficiently fast, such that all series are absolutely convergent and lead to continuous kernels.

However, as in the $\mathbb{R}^d$ case, such general kernels are hardly used. Instead, radial or zonal kernels are employed, following Guideline 2.10.

Definition 9.1. A kernel $K : S^{d-1} \times S^{d-1}$ is called *radial* or *zonal* if $K(x,y) = \varphi(\mathrm{dist}(x,y)) = \psi(x^T y)$ with univariate functions φ, ψ and the geodesic distance $\mathrm{dist}(x,y) = \arccos(x^T y)$. The function ψ is sometimes called the *shape function* of the kernel K.

Suppose that $a_{\ell,k} = a_\ell$, $1 \le k \le N(d,\ell)$. Then, by the addition theorem (9.1), we have

$$K(x,y) = \sum_{\ell=0}^{\infty} \tfrac{a_\ell N(d,\ell)}{\omega_{d-1}} P_\ell(d; x^T y) =: \sum_{\ell=0}^{\infty} b_\ell P_\ell(d; x^T y),$$

which shows that K is radial or zonal. Conversely, if K is radial we can expand the shape function ψ using the orthogonal basis $P_\ell(d;\cdot)$ for $L_2[-1,1]$ to get

$$K(x,y) = \sum_{\ell=0}^{\infty} b_\ell P_\ell(d; x^T y).$$

The addition theorem and the uniqueness of the Fourier series suffice to prove the following result.

Theorem 9.2. A kernel K of the form (9.2) with sufficiently fast decaying coefficients is zonal if and only if $a_{\ell,k} = a_\ell$, $1 \leq k \leq N(d, \ell)$.

Obviously, a zonal kernel is positive definite if all coefficients b_ℓ are positive. Moreover, it is also necessary that all coefficients are nonnegative (Schoenberg 1942). However, it is not necessary that all coefficients are strictly positive (Chen, Menegatto and Sun 2003*b*). The authors derived the following characterization.

Theorem 9.3. In order that a zonal kernel Φ is positive definite on S^{d-1} with $d \geq 3$ it is necessary and sufficient that the set $K = \{k \in \mathbb{N}_0 : b_k > 0\}$ contains infinitely many odd and even numbers.

The condition in this theorem is no longer sufficient on the unit circle (Pinkus 2004).

A first and most intuitive example of zonal functions comes from the $\mathbb{R}^d$ case. Suppose $K = \phi(\|\cdot\|_2) : \mathbb{R}^d \to \mathbb{R}$ is a positive definite and radial function on $\mathbb{R}^d$. Since we have $\|x - y\|_2^2 = 2 - 2x^T y$ for $x, y \in S^{d-1}$, we see that the restriction of K to S^{d-1} has the representation $K(x - y) = \phi(\|x - y\|_2) = \phi(\sqrt{2 - 2x^T y})$, so that it is indeed a zonal function with shape function $\psi = \phi(\sqrt{2 - 2\cdot})$. This immediately gives access to a huge class of zonal kernels on the sphere which are explicitly known and avoid calculation of any series.

This also raises the question if there is a connection between the (radial) Fourier transform $\widehat{K}$ of the positive definite function on $\mathbb{R}^d$ and the Fourier coefficients a_ℓ of the zonal function

$$\psi(x^T y) = \sum_{\ell=0}^{\infty} a_\ell \tfrac{N(d,\ell)}{\omega_{d-1}} P_\ell(d; x^T y).$$

Interestingly, there is a direct connection, which also shows that almost all positive definite and radial kernels on $\mathbb{R}^d$ define positive definite zonal kernels on S^{d-1} and the Fourier coefficients of the latter are all positive.

Theorem 9.4. (Narcowich and Ward 2004, zu Castell and Filbir 2005) Let K be a positive definite radial function having a nonnegative Fourier transform $\widehat{K} \in L^1(\mathbb{R}^d)$, and let $\psi(x^T y) := K(x - y)|_{x,y \in S^{d-1}}$. For $\ell \geq 0$, we have that

$$a_\ell = \int_0^\infty t \widehat{K}(t) J_\nu^2(t)\, \mathrm{d}t, \quad \nu := \ell + \tfrac{n-1}{2}, \tag{9.3}$$

where J_ν is the order ν Bessel function of the first kind. Moreover, if $\widehat{K}$ is nontrivial, *i.e.*, positive on a set of nonzero measure, then $a_\ell > 0$ for all ℓ.

This result was later generalized to conditionally positive definite basis functions (Narcowich, Sun and Ward 2006).

9.3. Error analysis

As in the case of (conditionally) positive definite kernels on $\mathbb{R}^d$, error estimates were first derived in the context of the associated reproducing kernel Hilbert space (Jetter, Stöckler and Ward 1999, Golitschek and Light 2001, Morton and Neamtu 2002, Hubbert and Morton 2004). Later, results on *escaping* the native Hilbert space, *i.e.*, for target functions from a rougher function space came up (Narcowich and Ward 2004). However, the involved rougher function spaces were not given by standard Sobolev spaces.

Here, we want to mimic the situation of Theorem 7.8. To this end we have to introduce Sobolev spaces on the sphere, which can be written as

$$W_2^\tau(S^{d-1}) = \left\{ f \in L_2(S^{d-1}) : \sum_{\ell,m} (1+\ell^2)^\tau |\widehat{f}_{\ell,m}| < \infty \right\}.$$

Naturally, to provide error estimates, the fill distance and the separation radius have to be redefined using *geodesic* distance now. If this is done, then it is possible to show (Narcowich, Sun, Ward and Wendland 2005*a*) the following analogue to Theorem 7.8.

Theorem 9.5. Assume $\tau \geq \beta > (d-1)/2$ and let ψ generate $W_2^\tau(S^{d-1})$ as its reproducing kernel Hilbert space. Given a target function $f \in W_2^\beta(S^{d-1})$ and a set of discrete points $X \subseteq S^{d-1}$ with mesh norm h_X, separation radius q_X and mesh ratio $\rho_X = h_X/q_X$, the error between f and its interpolant $s_{f,X}$ can be bounded by

$$\|f - s_{f,X}\|_{W_2^\mu(S^{d-1})} \leq C \rho_X^{\beta-\mu} h_X^{\beta-\mu} \|f\|_{W_2^\beta(S^{d-1})} \tag{9.4}$$

for all $0 \leq \mu \leq \beta$.

Note that (9.4) reduces to the expected error estimates when the approximation order is dictated by the rougher Sobolev space and if quasi-uniform data sets are considered. Finally, we should remark that a zonal function ψ generates $W_2^\tau(S^{d-1})$ if its coefficients a_ℓ in Theorem 9.2 decay like $\ell^{-2\tau}$.

10. Applications of kernel interpolation

Here, we review some practical application areas for kernel techniques which fit neither into Section 11 on machine learning nor into the final sections on solving partial differential equations. These techniques perform generalized interpolation of smooth functions using unstructured data. The background was described in Section 3 on optimal recovery, with conditional positive definiteness added from Section 6. Finally, special techniques for handling large-scale problems from Section 8 will occur at certain places. We group the applications by certain features that are sufficiently general to enable the reader to insert new applications into the right context. Unfortunately, our references cannot cover the application areas properly.

10.1. Modelling nonlinear transformations

Recovery problems like in (3.5) can of course be made vector-valued, and then they provide nonlinear multivariate mappings $F : \mathbb{R}^d \to \mathbb{R}^n$ with specified features expressible as linear conditions. Typical examples are *warping* and *morphing*. Warping is done by a fixed map taking an object of $\mathbb{R}^n$ to another object in $\mathbb{R}^n$, while morphing requires a parametrized scale of warping maps that describe all intermediate transformations. For these transformations, some input and output points have fixed prescribed locations, *e.g.*, keeping eyeballs fixed when morphing two faces, and these conditions take the form (3.1) or (3.3). Since kernel-based interpolation allows any kind of unstructured data, it is very easy to generate a warping or morphing map F with such conditions in any space dimension (Noh, Fidaleo and Neumann 2000, Glaunés, Vaillant and Miller 2004). However, the most popular applications (Gomes, Darsa, Costa and Velho 1998) avoid solving a linear system and prefer simple local techniques. Here is an open research field.

10.2. Exotic data functionals

This application area uses the fact that kernel techniques can recover functions from very general kinds of 'data' which need not be structured in any way. Any linear functional λ_j acting on multivariate functions is allowed in (3.3), provided that the kernel K is chosen to be sufficiently smooth to make $\lambda_j^x \lambda_j^y K(x,y)$ meaningful.

Guideline 10.1. Kernel methods can handle generalized recovery problems when the data are given by rather exotic linear functionals.

A typical example (Iske and Sonar 1996, Sonar 1996, Cecil, Qian and Osher 2004, Wendland 2005*a*) concerns postprocessing the output of *finite volume methods*. These calculate a set of values f_j of an unknown function f which are not evaluations of f at certain nodes x_j, but rather integrals of f over a certain small 'volume' V_j. Thus the functionals in (3.3) are

$$\lambda_j(f) := \int_{V_j} f(t)\,\mathrm{d}t, \quad 1 \le j \le N.$$

Usually, the domains V_j form a non-overlapping decomposition of a domain Ω. Then any recovery $\tilde{f}$ of f along the lines of Sections 3 and 6 will have the same local integrals as f, and also the global integral of f is reproduced. Thus postprocessing a finite-volume calculation produces a smooth function with correct local ‘finite volumes’. These functions can then be used for further postprocessing, *e.g.*, calculation of gradients, pressure, or contours.

This technique can be used in quite a general fashion. In fact, one can always add interpolation conditions of the above type to any other recovery problem, and the result will have the required conservation property.

Guideline 10.2. Within kernel-based reconstruction methods, it is possible to maintain conservation laws.

In some sense, morphing also maintains some kind of conservation.

Another similar case occurs when a certain algorithm produces an output function which does not have enough smoothness to be the input of a subsequent algorithm. An intermediate kernel-based interpolation will help.

Guideline 10.3. Using kernel-based techniques, we can replace a non-smooth function by a smooth one, preserving any finite number of data which are expressible via linear functionals.

We stated this in the context of conservation here, but it will occur again later with a different focus.

A somewhat more exotic case is the recovery of functions f from *orbital derivatives* along trajectories $X(t) \in \mathbb{R}^d$ of a dynamical system (Giesl 2005). The data at x_j are not $f(x_j)$ but the derivative of $t \mapsto f(X(t))$ at t_j of the trajectory passing through $x_j = X(t_j)$. This information, when plugged into a suitable recovery problem, can be used to prove stability of solutions of dynamical systems numerically, by constructing Lyapunov functions as solutions to recovery problems from unstructured orbital derivative data.

10.3. Recovery from many scattered values

The recovery of a multivariate function f from large samples of unstructured data $(x_j, f(x_j))$, $1 \le j \le N$ on a domain $\Omega \subseteq \mathbb{R}^d$ theoretically follows the outlines given in Sections 2 and 3. However, for large N there are specific problems that need special numerical techniques of Section 8. We do not repeat these here. Instead, we focus on *terrain modelling* as a typical application.

As long as terrains are modelled as elevations $z = f(x)$ described by a bivariate function f and using gridded data $(x_j, z_j) = (x_j, f(x_j)) \in \mathbb{R}^3$ as in current geographic databases (*e.g.*, the US Geological Survey), there are no serious problems. But the raw elevation data often come in an irregular distribution, because they are sampled along routes of ships, aeroplanes, or satellites. This means that the fill distance (7.6) will be much larger than the separation radius (7.7). The latter is given by the sampling rate along each route, while the first depends on how well the routes cover the domain. The problem data live on two different scales: a smaller one along the sampling trajectories and a larger one 'between' the trajectories. Section 7 tells us that the recovery error is dominated by the fill distance, while the condition is determined by the separation radius. This calls for *multiscale* methods, which are also necessary in many other applications in geometric modelling.

Multiscale techniques, as described in Section 8, use Guideline 8.4, but they have to split the given large dataset X into a nested sequence (8.5). Each subset X_j of the data should be quasi-uniform in the sense of (7.8). This can be done by sophisticated *thinning algorithms* (Floater and Iske 1998) and using kernels of different scales at different levels. For details, we refer the reader to recent books covering this subject (Iske 2004, Dodgson, Floater and Sabin 2004). A promising new approach via multiscale kernels (Opfer 2006) directly resolves such problems on several scales, but work is still in progress.

10.4. Recovery of implicit surfaces

This is different from the previous case, because the resulting surface should not be in *explicit* form $z = f(x)$ with $x \in \Omega \subset \mathbb{R}^2$. Instead, the goal is to find an *implicit* description of a surface as the level set $\{x \in \mathbb{R}^3 \ : \ g(x) = 0\}$ of a scalar function $g \ : \ \mathbb{R}^3 \to \mathbb{R}$. The given data consist of a large set of unstructured points $x_j \in \mathbb{R}^3$, $1 \leq j \leq N$ expected to lie on the surface, *i.e.*, to satisfy $g(x_j) = 0$ for all j in question. This is an important problem of *reverse engineering*, if the data come from a laser scan of a 3D object. The final goal is to produce an explicit piecewise CAD-compatible representation of the object from the implicit representation.

The basic trick for handling such problems is to view them as a plain interpolation problem for g with values 0 at the x_j. To avoid the trivial function a number of points 'outside' the object has to be added with values less than zero and points 'inside' with values larger than zero.

To this end, it is assumed that the surface indeed divides $\mathbb{R}^3$ into an inner and outer part, meaning that the surface is closed and orientable and has a well-defined outer normal vector at each point. With these additional assumptions at hand, the first task is to find outer normal vectors for each point. This can be done by using additional information, such as the position of the laser scanner, or by trying to fit in each point a local tangent plane to the surface. In the latter case, the so-calculated normals have to be oriented consistently, which is, unfortunately, an NP-hard problem. However, there exist good algorithms producing in most cases a satisfactory orientation (Hoppe, DeRose, Duchamp, McDonald and Stuetzle 1992, Hoppe 1994).

With these normals at hand, the additional points can be inserted along the normals. A function value which is proportional to the signed distance to the surface is assigned to each new point, making the interpolation problem nontrivial.

However, this procedure might triple the often already huge number of data sites, such that efficient algorithms, like those described in Section 8, are required. For example, this has been successfully demonstrated in various papers (Carr, Beatson, Cherrie, Mitchell, Fright, McCallum and Evans 2001, Turk and O'Brien 2002, Ohtake, Belyaev, Alexa, Turk and Seidel 2003*a*, Ohtake, Belyaev and Seidel 2003*b*) and is already well established in industry.[3]

10.5. Transition between different representations

Consider two different black-box numerical programs which have to be linked together, in the sense that the first produces multivariate discrete output data describing a function f while the second program needs different data of f as its input. This occurs if two FEM programs with different meshes and elements are used, or if results of a program need some post-processing.

Everything is fine if the two programs use function representations based on the same discrete data. Otherwise, an intermediate kernel-based recovery will be useful. The output data of the first program is taken as input of a kernel-based recovery process to find a function $\tilde{f}$ close to f. Then the input data for the second program is derived from $\tilde{f}$.

A typical field of this application is aeroelasticity. Here, the interaction between the flow field around an elastic aircraft during flight and the aircraft itself is studied. A deforming aircraft leads to more realistic lift and drag and, particularly in the design of large aircrafts, has to be taken into account.

The black-box solvers involved are the aerodynamic solver for the computation of the flow field and a structural solver for the computation of the deformation of the aircraft. While the flow field is often discretized using high-resolution finite volume methods in Eulerian coordinates, the structure of the aircraft is generally described by a coarse finite element discretization in Lagrangian coordinates. The exchange of information is limited to transfer forces from the aerodynamic program to the structural mesh and displacements from the structural mesh to the aerodynamical one. In particular the latter can be modelled as a scattered data interpolation problem. This has been done successfully, for example, in a series of papers (Farhat and Lesoinne 1998, Beckert 2000, Beckert and Wendland 2001, Ahrem, Beckert and Wendland 2005) and is already on its way to become an industry standard. The exchange of forces is in general differently achieved such that the sum of all forces and the virtual work are conserved between both models.

[3] `http://www.farfieldtechnology.com/`
`http://aranz.com/research/`

Interestingly, an early application (Harder and Desmarais 1972) in aircraft engineering is the first paper in which thin-plate or surface splines were used in a scattered data interpolation problem, while the theory arrived four years later (Duchon 1976).

11. Kernels in machine learning

The older literature on radial basis functions was dominated by applications in *neural networks*, in which sigmoid response functions were gradually replaced by radial basis functions over the years. Many papers of this kind call a function (3.4) with a radial kernel a *radial basis function network.* We do not want to explain this machinery in detail here, because *kernels* provide a much more general and flexible technique replacing classical neural networks in learning algorithms. There are close connections of machine learning to pattern recognition and data mining, but we have to be brief here and prefer to focus on learning, leaving details to standard books on machine learning with kernels (Schölkopf and Smola 2002, Shawe-Taylor and Cristianini 2004).

11.1. Problems in machine learning

We start with an introduction to the notions of machine learning, based on the recovery problems in Section 3. These are subsumed under *supervised learning*, because the expected response y_j to an input $x_j \in \Omega$ is provided by the unknown 'supervisor' function $f \; : \; \Omega \to \mathbb{R}$. If the target data y_j can take non-discrete real values, the supervised recovery problem is called *regression*, while the case of discrete values is called *classification.* In the latter case the input set Ω is divided into the equivalence classes defined by the different target values. After learning, the resulting function $\tilde{f} \approx f$ should be able to classify arbitrary inputs $x \in \Omega$ by assigning one of the finitely many possible target values. For instance, a classification between 'good' and 'bad' inputs x_j^+ and x_j^- can be done by finding a hyperplane in feature space which separates the features $\Phi(x_j^+)$ and $\Phi(x_j^-)$ of 'good' and 'bad' inputs in a best possible way. This can be done by linear algebra or linear optimization, and is an instance of Guideline 2.2.

In many applications, classification is reduced to regression by:

(1) embedding the discrete target values into the real numbers,
(2) solving the resulting regression problem by some function $\tilde{f}$,
(3) classifying new inputs x by assigning the discrete target value closest to $\tilde{f}(x)$.

Thus we shall focus on regression problems later, ignoring special techniques for classification.

Unsupervised learning has inputs $x_j \in \Omega$ but no given target responses y_j associated to them. The goal for learning is given semantically instead. A frequent case is *clustering*, which is classification with just a few target values whose calculation is part of the problem. Another unsupervised technique is the determination of anomalies, outliers, or novelties. This can be seen as a classification where only the 'normal' inputs are known beforehand, while future 'abnormal' inputs have to be detected. A more general topic closely related to unsupervised learning is *data mining*, which attempts to discover unknown relations between given data (Hastie, Tibshirani and Friedman 2001), but we cannot go into details.

11.2. Linear algebra methods in feature space

Many pattern recognition or learning techniques apply a linear algebra technique in feature space, and thus they use Guideline 2.2. Since the kernel matrix contains all geometric information on the learning sample, the algorithms are based on the kernel matrix or on information derived from it. A simple novelty detection could, for instance, just check how far a new feature vector $\Phi(x)$ is away from the mean of the 'normal' feature vectors $\Phi(x_j)$ and declare it 'abnormal' if it is 'too far away'. Of course, there are statistical background arguments to support certain decision rules.

Primitive binary classification can take the means μ^+ and μ^- of the feature vectors $\Phi(x_j^+)$ and $\Phi(x_j^-)$ of the 'good' samples x_j^+ and 'bad' samples x_k^-, and then classify a new input x by checking whether $\Phi(x)$ is closer to μ^+ or μ^-. Of course, there are more sophisticated methods with statistical foundations, but the upshot is that a kernel defined via a feature map is all that is needed to start a linear algebra machinery, ending up with certain statistical decision rules.

A very important background technique for many pattern recognition and learning algorithms is to attempt a *complexity reduction* of the input data first. If this is possible, anomalies can be detected if they do not fit properly into the reduction pattern for the 'normal' data. The most widely used method for complexity reduction proceeds via *principal component analysis*, which in the case of kernel-based methods boils down to a singular-value decomposition of the kernel matrix followed by projection onto the eigenspaces associated to large singular values.

11.3. Optimization methods in feature space

But the most important numerical methods in machine learning are *optimizations*, not linear algebra techniques. For illustration, we take a closer look at unsupervised learning in the regression case, which in Section 3 was called a recovery problem. The *reproduction–generalization* dilemma stated in Guideline 3.3 is observed in machine learning by minimizing both a *loss function* penalizing the reproduction error and a *regularization* term penalizing instability and assuring generalization. These two penalty terms arise in various forms and under various assumptions, deterministic and nondeterministic, and they can be balanced by taking a weighted sum as an objective function for joint minimization. A typical deterministic example is (7.13) summing a least-squares loss function and a native space norm penalty term. Another case is the sup-norm loss function

$$\epsilon := \max_j |y_j - f(x_j)|$$

arising indirectly in (3.9) and added to the native space norm to define the objective function $\frac{1}{2}\|f\|_{\mathcal{K}}^2 + C\epsilon$ to be minimized.

Both cases, like many others in machine learning, boil down to quadratic optimization, because (2.6) allows explicit and efficient calculation of the native space norm on the trial space (2.3) via the kernel matrix defined for the training data. This applies to all techniques using the quadratic penalty

$$\alpha \in \mathbb{R}^N \mapsto \alpha^T A_{K,X} \alpha = \|f\|_{\mathcal{K}}^2 \tag{11.1}$$

to guarantee stability and generalization. For large training samples, the resulting quadratic programming problems have to cope with huge positive definite kernel matrices in their objective function, calling for various additional numerical techniques like principal component analysis to keep the complexity under control. Of course one can also get away with linear optimization if the quadratic term is replaced by minimization of terms like $\|A_{K,X}\alpha\|_\infty$ or $\|\sqrt{A_{K,X}}\alpha\|_\infty$ with a similar penalty effect. Again, the kernel matrix is the essential ingredient.

But this technique is not limited to learning algorithms. One can use it for regularizing many other methods, because one has a cheap grip on high derivatives.

Guideline 11.1. Quadratic penalty terms (11.1) using the square of the native space norm of a kernel-based trial function are convenient for regularizing ill-posed problems.

Since this only requires the trial space to consist of translates of a single positive definite kernel, and since such trial spaces have good approximation properties, kernel-based methods are good candidates for solving ill-posed and inverse problems (Lewitt, Matej and Herman 1997, Hon and Wu 2000*b*, Cheng, Hon, Wei and Yamamoto 2001*b*, Cheng, Hon and Yamamoto 2001*a*, Green 2002, Hon and Wei 2002, 2003, 2005). Solving ill-posed and inverse problems by kernel techniques has a promising future.

11.4. Loss functions

After looking at the penalty for instability, we have to focus on the *loss* function, while we assume an at least quadratic optimization using (11.1) as part of the objective function. There are various ways to define loss, but they have seriously different consequences, not only from a statistical, but also from a numerical viewpoint. We ignore the vast literature on statistical learning theory here and focus on computationally relevant questions with implications for other kernel-based techniques.

The quadratic least-squares loss in (7.13) has the consequence to add a constant diagonal to the kernel matrix. This is the old Levenberg–Marquardt regularization of least-squares problems, but it has the disadvantage that the solution will not have a reduced complexity. The resulting coefficient vector $\alpha \in \mathbb{R}^N$ for N training samples will not necessarily have many zeros, so that the kernel-based model (3.4) has full $\mathcal{O}(N)$ complexity.

On the other hand, Guidelines 3.16 and 3.17 tell us that a complexity reduction should be possible, using only $n \ll N$ terms in the solution (3.4). This is achieved by using *linear* loss constraints like (3.9) instead of quadratic ones. Then the Kuhn–Tucker theory restricts the optimal solution via the *active* constraints. In the literature on machine learning, this is the *support vector machine* philosophy, because the feature vectors $\Phi(x_j)$ for the 'active' indices j with $|f(x_j) - y_j| = \epsilon$ are called 'support vectors' for some reason or other.

Guideline 11.2. Complexity reduction via linear loss constraints is useful for most recovery situations, deterministic or non-deterministic.

Since many numerical methods can be reformulated as recovery problems, this has an unexpectedly wide range of possible applications. We use it for adaptive meshless collocation methods in Section 14. There are good chances that future methods for PDE solving will take the form of adaptive optimization routines with linear loss constraints leading to complexity reduction.

11.5. Kernels in learning theory

Theoretical research on learning has close connections to approximation theory, and it is naturally focusing on kernels (Smola and Schölkopf 1998, Cucker and Smale 2001, Schölkopf and Smola 2002, Zhou 2002, Smale and Zhou 2003, Poggio and Smale 2003). Most of this is based on statistical learning theory. Since we want to stay on the numerical analysis side, we only present the most important connection to approximation by kernels.

A central question in supervised learning is to have bounds for the necessary number N of training data (x_j, y_j) to guarantee the availability of a trained model $\tilde{f}$, based on these data, which has a small generalization error $\|f-\tilde{f}\|_\Omega \le \epsilon$ in some norm $\|\cdot\|_\Omega$ over the input domain Ω. This problem can be handled using Theorem 7.8. In particular, if the true model f lies in some Sobolev space $W_2^\beta(\Omega)$ containing the native space for our kernel, and if X is a quasi-uniform sample set of N points in Ω with fill distance $h_{X,\Omega}$, we can find an exact data reproduction $s_{f,X}$ based on X such that Theorem 7.8 provides an error bound of order $h_{X,\Omega}^{\beta-\mu}\|f\|_{W_2^\beta(\Omega)}$ for the generalization error in the Sobolev norm $\|\cdot\|_{W_2^\mu}(\Omega)$. Thus the necessary number $N \approx h_{X,\Omega}^{-1/d}$ of training samples to handle all nonzero unknown functions $f \in W_2^\beta(\Omega)$ to an error $\|f-\tilde{f}\|_{W_2^\mu(\Omega)} \le \epsilon$ behaves like

$$N \ge C \cdot \left(\frac{\epsilon}{\|f\|_{W_2^\beta(\Omega)}}\right)^{\frac{-d}{\beta-\mu}}$$

for $0 \le \mu < \beta$. Guideline 3.13 arises here again, because smoothness of the kernel and the model pays off. There are similar bounds in other norms, but we do not go into details. Unfortunately, there are no deterministic results yet which support Guideline 3.16 in a quantitative way, reducing N if the reproduction quality is relaxed.

12. Meshless methods

Here, we start considering applications of kernels within methods solving partial differential equations. These are published in abundance, mainly in journals focusing on computational techniques in engineering and sciences, and this paper should help the user to sort them out properly. To this end, we derive some guidelines for using kernels in numerical methods, but this will need some general considerations first. To set the stage properly, we recall the fundamental dichotomies between

- strong and weak problem formulations
- test and trial functions
- stationary and nonstationary scales of trial spaces
- implicit or explicit shape functions
- symmetric and unsymmetric methods

and consider

- regularity of solutions
- consistency, *i.e.*, reproduction of polynomials
- adaptivity
- necessity of global spatial discretizations
- numerical integration.

These issues are intimately related, as we shall see.

12.1. Strong and weak problems

Strong problems define solutions as functions satisfying a partial differential equation and certain boundary conditions *pointwise*, employing evaluations of functions and classical derivatives. *Weak problems* replace point evaluations by local integrations against *test functions* or (weak) derivatives thereof, introducing numerical integrations. Both apply 'tests' to check whether a 'trial' function is a solution. Their difference is not on the 'trial' side, but on the 'test' side. We shall come back to this later.

Strong methods can be called 'integration-free', and this is often more important than the notion of 'mesh-free' or 'meshless'. As far as point evaluations are concerned, there is no big difference between weak and strong methods, since the weak methods also use strong function values for their integration routines. The crucial point of weak formulations, however, is to apply integration by parts to the integrals of derivatives against test functions, thus reducing the necessary order of differentiability and allowing Hilbert space methods like Dirichlet's principle.

Strong formulations imply stronger regularity assumptions, *i.e.*, classical differentiability with Hölder continuity of the highest derivatives occurring in the differential equations. Weak formulations can get away with lower regularity and lower-order derivatives, but the derivatives are not classical ones. While this argument is independent of numerical methods, regularity is closely connected to them, since convergence orders usually increase with regularity.

Guideline 12.1. If the PDE problem has a rather regular solution, the user should apply techniques that make use of this regularity, and can choose between weak and strong problem formulations. If the solution will definitely have low regularity, the user should first try to convert the problem to another with more regularity, *e.g.*, by giving expected singularities or discontinuities a special treatment. If the final problem still leads to a solution with low regularity, the user is forced to pose a weak problem, but must expect poor numerical performance of any numerical method.

If there is enough regularity to have a choice between weak and strong problems, the connection of the problem formulation to numerical integration becomes important. Weak formulations introduce additional numerical integrations which are not necessary for strong formulations. These numerical integrations increase the algorithmic complexity and introduce a possibly avoidable source of numerical errors.

Guideline 12.2. Strong problem formulations avoid certain numerical integrations, but they have to assume higher regularity than weak formulations.

The integration error can be quite serious (Ciarlet 1991, Babuška *et al.* 2002) and needs a careful selection of integration techniques. In particular, if regularity is high to allow high-order methods like *h-p* finite elements in a weak formulation, the integration quality must be increased properly to adjust to the convergence order, so that the final error is not dominated by the one induced by numerical integration. This makes it questionable to go for a weak problem formulation in the case of high regularity, because strong formulations without integrations become an option in that case.

12.2. Trial functions

If we rule out purely discrete techniques like plain finite differences, the approximate solutions of partial differential equations are usually represented as linear combinations of *trial functions*. These come in a great variety, *e.g.*, as polynomials, piecewise polynomials (splines, box splines, or finite elements), shape functions, particle functions, generalized finite elements, wavelets, or kernel translates. Furthermore, they do not come singly, but usually as a whole *scale* of spaces, and then the question of *stationary* or *nonstationary* scaling comes up as in Section 7. Let us have a closer look at trial spaces in general in order to see where kernels are useful.

Guideline 12.3. Trial functions should

- provide a good approximation to the solution,
- be effectively evaluable,
- easy to modify, and
- easy to integrate numerically, in the case of weak problems.

They should only in the latter situation be dependent on the test side. We shall now look at these properties one by one, starting with approximation properties.

In many cases, *e.g.*, for finite elements, scales of trial spaces attain their approximation power via a *geometric domain discretization* of the underlying domain up to some granularity h describing something like the maximum diameter of a local polyhedral support of a trial function. Certain methods using shape functions or translated kernels do not split the domain geometrically, but use a cloud of points that 'fills' the domain so that h is a *fill distance* such as (7.6), which measures the radius of the largest ball with centre in the domain but without one of the data points. In both cases, there is a *domain discretization* involved.

But as far as approximation power is concerned, it is by no means mandatory that a scale of trial spaces requires a geometric global domain discretization of any kind.

Guideline 12.4. If the expected solution of a problem has a good approximation from a low-dimensional space of global functions, the trial space should be selected accordingly, without discretizing the domain at all. If singularities of known form and place are to be expected, they should be included into the trial space, no matter what the actual numerical method is.

Note that a possibly missing space discretization for the trial space is just one aspect when looking at 'meshless methods'. There may be integration nodes in certain cases, and there may be a space discretization for the test side which we have not yet looked at. Currently, most meshless methods are still using global space discretizations, but allow us to add adaptive local refinement when necessary. However, the user should keep in mind that spectral methods (Fornberg and Sloan 1994) or general trial spaces without space discretization are to be considered as alternatives when the expected properties of the solution allow them.

Guideline 12.5. High approximation *orders* are not related to domain discretization, but to smoothness. They are achievable if the solution of the problem is sufficiently smooth. This is independent of the trial space. But they also require a trial space that can make use of that smoothness.

Such spaces must have higher smoothness themselves, as in the p-version of the finite element method. A trial space with good approximation properties should thus have p-adaptivity in the sense that it guarantees the highest possible approximation order attainable for the (unknown) smoothness of the solution. By Section 7 and Guideline 7.9 we know that nonstationary scales of kernel-based trial spaces have both a p- and h-adaptivity, but theory still requires a space discretization with a small fill distance h, because it focuses on a worst-case scenario. It is a future challenge to provide a sound mathematical basis for data-dependent h-type adaptivity such as the support vector technology within machine learning. Future adaptive optimization strategies for PDE solving should use Guideline 3.18 and select spatial resolutions locally where needed, and automatically yield optimal local approximation orders depending on the local smoothness of the solution.

Some applications require good approximations of higher derivatives of the solution, *e.g.*, if pressure or stress is to be evaluated from displacements ion mechanics. This calls for smooth trial functions.

Guideline 12.6. Because the node connectivity problems of piecewise polynomials increase dramatically with smoothness requirements and space dimension, it is much easier for meshless kernel-based methods than for finite elements to generate smooth trial spaces, in particular for higher space dimensions.

Standard results concerning numerical methods for solving ODEs and time-dependent PDEs suggest that good convergence orders are obtained by high-*consistency* orders, provided that *stability* is satisfied. This is not directly related to the approximation power of trial spaces. Unfortunately, *consistency* occurs in quite a number of application papers on meshless methods in a nonstandard way, and we shall later describe its questionable use there.

We now leave approximation quality and focus on evaluation efficiency of trial functions. Though not standard in the literature, we distinguish between *explicit* and *implicit* evaluation of trial functions. For *explicit* evaluation, there is a simple formula, *e.g.*, $\exp(-0.3 * \|x - x_j\|_2^2)$, for each trial function, and there is no need to look up a number of other nodes or to evaluate geometric data. This is the standard technique for kernel-based trial spaces. *Implicit* evaluation means that each trial function value is the result of a subroutine call to a function that depends on multiple data in a somewhat complex and geometry-dependent way. This applies to finite elements and all 'shape functions' which are the result of pointwise local optimizations like *moving least squares* of Section 7.

Guideline 12.7. If applications need to evaluate the solution on extremely many points, implicit trial spaces may not be the best choice.

It often happens that the calculation of the parameters of the representation of a solution is faster than the generation of all values needed for postprocessing, *e.g.*, for visualization. Then *evaluation* becomes more important than *solving*. *A posteriori* display of a scattered-data interpolant to the actual solution along the lines of Section 7 is always possible, of course, but it is a problem of its own and induces additional errors.

Another efficiency argument arises when the dimension of the trial space is large. This should be avoided following Guideline 12.4, but it always occurs if the trial space is using a space discretization with fine granularity. Even if there is not too much connectivity between geometric information, *i.e.*, if the method is meshless, we need to have a fast method for range queries retrieving neighbours of nodes. Similar problems always come up when trial spaces need some *localization*. There are various ways to cope with it, *e.g.*, wavelets, multipole, partition-of-unity, and multilevel methods, but they all seem to be closely connected to the choice of a useful basis, either *a priori* or adaptively. This brings us to the next issue: the adaptivity properties of trial spaces.

The really serious situations for the choice of the trial space occur when singularities will arise, but at places not known beforehand. This is the case for certain fluid dynamics, advection-diffusion, or crack propagation problems. However, it does not make sense to use a fine global space discretization when there will be just a local effect that calls for a finer local resolution. This is observed by plenty of *adaptive* methods. They sometimes just re-mesh a global space discretization locally where necessary, or they add new and more flexible elements into the fixed basic triangulation, but both of these tasks are not easy. Particle- or kernel-based methods using clouds of scattered points can adapt by adding or deleting points where necessary, but they usually do not need to update geometric contingency information that arises with meshes or triangulations. This is the punchline when *meshless methods* are characterized (Belytschko *et al.* 1996*b*) as *constructing the approximation entirely in terms of nodes.* The cited article considers meshless approximations based on

- moving least squares
- kernels
- partitions of unity

and states that these *three methods are in most cases identical except for the important fact that partitions of unity enable p-adaptivity to be achieved.* Furthermore, kernels occur in all three, and this is another reason why kernels are a central tool in meshless methods. Some authors even talk of *truly meshless methods* when they want to stress that they do not need numerical integration, but we suggest stating precisely to which extent spatial discretizations need to be maintained, and whether the trial functions can be accessed explicitly or implicitly.

12.3. Kernel-based trial spaces

At this point, we should show how 'representability in terms of nodes' is understood in meshless methods and how it is related to kernel-based trial functions, establishing a very close connection of nearly all meshless methods to kernels. The idea of 'nodes' is roughly the same as the 'centres' for standard kernel approximations as in Section 7. In the simplest case, the trial space should be spanned by multivariate functions $\varphi_i(x, x_i),\ i \in I$, which are functions of $x \in \mathbb{R}^d$ depending on a single 'node' or 'centre' or 'particle position' $x_i \in \mathbb{R}^d$. This function can be seen as a 'smoothed particle' as in *smoothed particle hydrodynamics*, SPH (Monaghan 2005), and it is called *shape function* or *particle function* in the literature. For a meshless method, there should be no complicated geometric connection between nodes like a triangulation of the convex hull of the nodes with the nodes as vertices (this could then be called a 'mesh'). It should be easy to extend the trial space by adding some new nodes and associated trial functions (this is called 'h-adaptivity' in FEM terms) without updating the connectivity information. In this sense, meshless methods can be seen as an alternative to adaptive finite element methods.

For many reasons, the functions $\varphi_i(x, x_i)$ in meshless methods should be

- translation-invariant and
- compactly supported around the node x_i.

This implies that they should necessarily have the form

$$\varphi_i(x, x_i) := K\bigl(x - x_i\bigr), \quad i \in I \tag{12.1}$$

with a compactly supported translation-invariant kernel K of small support, provided that they depend on no other neighbouring node.

Guideline 12.8. Translation-invariant trial functions for meshless methods are always kernel-based, if they are dependent on a single node.

This implies that trial spaces spanned by functions of the form (3.4) occur canonically in meshless methods, and the previous sections have accumulated much information on those spaces.

But the literature on meshless methods also uses 'shape functions' defined *implicitly* via local processes such as moving least squares. Then the resulting trial functions depend on more than one node, though this is often ignored in the notation. In fact, for each node x_i there is a trial function φ_i depending on x_i and some of its neighbours, if they fall within the support of the weight function associated to the node x_i. Kernels occur here only via the weight functions used, and they need not be positive definite. For scattered nodes, the resulting trial functions will not be translation-invariant.

12.4. Reproduction of polynomials

For MLS-based shape functions, we know from Section 7 that polynomial reproduction

$$\sum_{i \in I} p(x_i)\varphi_i(x) = p(x) \quad \text{for all } x \in \mathbb{R}^d,\ p \in \pi_m(\mathbb{R}^d) \tag{12.2}$$

can be achieved under mild additional assumptions, where $\pi_m(\mathbb{R}^d)$ stands for the space of d-variate polynomials of degree at most m. Note that the partition of unity property (7.17) coincides with polynomial reproduction of degree zero.

In application papers, polynomial reproduction properties are often called *consistency conditions* (Belytschko *et al.* 1996*b*), and very many papers seem to understand *reproducing kernels* via the above reproduction property, not via (2.9) in Hilbert spaces. Some also seem to assume that convergence follows as soon as there is a consistency condition of some nonnegative order in the above sense, but this argument has no solid foundation, since the usual Lax-type theory understands consistency differently and is modelled for discretizations of time-dependent problems. Mathematicians will find plenty of open questions concerning convergence and error bounds of meshless methods, while many engineers seem to believe themselves to be on solid ground once they have what they call consistency.

Sometimes the notion of *completeness* is used in the sense of *convergent approximate polynomial reproduction*, *i.e.*, (12.2) holding for $h \to 0$. In particular, *linear completeness* often means convergent approximate reproduction of linear functions (Belytschko *et al.* 1996*b*). This is different from the usual notion of completeness in mathematics, and it must be used with extreme care, in particular when assuming that it implies convergent approximate reproduction of *piecewise* linear functions.

Guideline 12.9. Within meshless methods, the notions of *consistency* and *completeness* should be used with caution.

Anyway, the polynomial reproduction property (12.2) appears in many meshless methods. In fact, recent surveys (Li and Liu 2002, 2004, Fries and Matthies 2004) of meshless methods focus entirely on methods with exact polynomial reproduction. However, it must be stated clearly that exact polynomial reproduction is not necessary for convergence, as is shown, for example, by the rigorous convergence analysis of the generalized finite element method (Babuška *et al.* 2003), and the symmetric (Franke and Schaback 1998*b*) and unsymmetric (Schaback 2005*b*) meshless collocation methods. Polynomial reproduction appears to be popular because it is necessary in convergence arguments for *stationary* scales of trial functions, using Strang–Fix conditions or the Bramble–Hilbert lemma. But it is not mandatory to use these tools. By Theorem 7.8, optimal approximation orders in Sobolev spaces are attained without it in very general situations, not only for interpolation from nonstationary scales of kernel-based trial spaces.

In view of these remarks, future work should remove exact polynomial reproduction from the assumptions of many meshless methods. Instead, care must be taken to conserve physical properties like mass and momentum in applications. This is only loosely related to polynomial reproduction.

12.5. Particle methods

After this detour into polynomial reproduction we still have to look at a class of methods that arrives at meshless trial spaces via a slightly different approach. *Smoothed particle hydrodynamics* (SPH) use spatial kernel approximations that we called *discretized kernel convolutions* in Section 7. This means that a suitably scaled and normalized kernel K is chosen such that (7.1) holds, and a discretization of the convolution integral implies

$$(K * f)(x) \approx \sum_{i \in I} w_i K(x, x_i) f(x_i) \quad \text{for all } x \in \mathbb{R}^d$$

with integration weights w_i at integration nodes x_i. The linear unknowns here are $f(x_i)$, while the points x_i are interpreted as *particle positions* and can be considered as nonlinear parameters whose number and value can change. The name of the technique is derived from the fact that the right-hand side writes a function or vector field as a sum over the local kernel-controlled influences of discrete particles at the points x_i. Thus the logic of SPH does not directly aim at trial spaces, but rather parametrizes fields describing flows in the form (3.4) we had in the beginning, by using the right-hand side of the above approximation. All other operations, *e.g.*, setting up momentum equations, are performed using the parametrized flow. Since the background problems are time-dependent, the above spatial discretizations lead to large systems of ordinary differential equations, where time discretization is another issue we do not address here.

To achieve a good approximation in the continuous convolution error (7.1), Theorem 7.1 tells us that the kernel should reproduce low-order polynomials well, but not necessarily exactly. If the integration scheme is exact for low-order polynomials, and if the kernel convolution reproduces low-order polynomials exactly, this implies the partition-of-unity property for the trial functions $w_i K(\cdot, x_i)$, but there will be no exact reproduction of higher-order polynomials. This problem can be removed by dropping the philosophy of discretizing a convolution integral, and going radically over to functions (12.1) with exact or approximate polynomial reproduction. This is called the *reproducing kernel particle method* (RKPM), when the rest of the SPH is maintained, *i.e.*, when discretized systems are derived from parametrized kernel-based field representations some way or other. We refer the reader to a recent survey article (Li and Liu 2002) and a book (Li and Liu 2004) on SPH and RKPM techniques, containing long lists of references, and describing many variations induced by additional physical constraints. But remember that meshless methods based on stationary moving least squares, reproducing kernels, or partitions of unity are *in most cases identical* (Belytschko *et al.* 1996*b*), so that all variants have to be looked at carefully.

12.6. Residuals, test functionals and functions

After considering the trial side, we should now focus on the test side. If we assume that the trial side has somehow produced some trial function which is a candidate for an approximate solution of the partial differential equation and the boundary conditions, we want to conclude that this trial function is close to the real solution. This is the job of the *test* side. In contrast to Guideline 12.4, we have the following rule, since the test side has to consider *security*.

Guideline 12.10. Space discretization is much more important on the test side than on the trial side.

But we postpone discretization on the test side for a while, noting that the above guideline calls for unsymmetric methods we deal with later.

If we rewrite the differential equation and the boundary conditions as differences $L(u) - f = 0$ which should be zero for the exact solution u, an approximate solution $\tilde{u}$ should make the *residuals* $L(\tilde{u}) - f$ small everywhere. Usually, to conclude that the error $u - \tilde{u}$ is small, it suffices to make sure that the residuals are small, because the solution of any well-posed linear problem will be continuously dependent on the data, implying

$$\tilde{u} - u \text{ small, if all residuals } L(\tilde{u}) - L(u) = L(\tilde{u}) - f \text{ are small,}$$

where 'all' means residuals of differential equation(s) and boundary condition(s) altogether, as many as are present in the problem. This means that 'testing' should usually make sure that the residuals are zero or at least small *globally*. Numerical techniques aiming at globally small residuals are often called *methods of weighted residuals.*

Guideline 12.11. Globally small residuals imply small errors for well-posed linear problems, *i.e.*, if the solution is continuously dependent on the data. But one must make sure that the notions of 'well-posedness' and 'globally small' are consistently defined.

In fact, if we pack differential equations and boundary conditions into one single linear operator $L : U \to F$, continuous dependence requires fixing spaces U and F for the solution u and the data f of the problem $L(u) = f$ such that

$$\|u\|_U \le C\|L(u)\|_F \tag{12.3}$$

holds, *i.e.*, L has a continuous inverse taking the data into a solution having these data. Then one must ensure that 'globally small' residuals for an approximate solution $\tilde{u}$ implies that the corresponding non-discrete norm $\|L(u) - L(\tilde{u})\|_F$ is also small, and *vice versa.* Thus, even when discretization of residuals is not an issue, the choice of a *residual norm* is important.

This is closely connected to the distinction between strong and weak problems. For strong problems, the residual norm is usually something like the $\|\cdot\|_\infty$ norm, while weak problems will use 'weaker' norms such as $\|\cdot\|_2$. But in most cases small residuals in the $\|\cdot\|_\infty$ norm will also be small in the $\|\cdot\|_2$ norm, so that, even if the $\|\cdot\|_2$ norm is the correct one for continuous dependence, users are safe if they minimize $\|\cdot\|_\infty$ instead, *i.e.*, solving a strong instead of a weak problem. This requires the trial space and the data f to have enough smoothness for $\|L(\tilde{u}) - f\|_\infty$ to be well defined, but this is usually not a big problem in many applications.

Guideline 12.12. If trial functions and data are smooth enough, users can often use a strong formulation even if a corresponding weak formulation is known to be well posed.

12.7. Global residual minimization

There is a natural class of numerical methods related to weighted residuals, *i.e.*, methods that globally optimize residuals in the correct residual norm. These will always lead to an optimization problem instead of a linear system, reminding us of Guideline 3.18 and the complexity-reducing optimization problems in Section 11 on machine learning. In the case of L_2 residual minimization, this is the well-known *method of* (*continuous*) *least squares*, and there the optimization problem is quadratic, boiling down again to a linear system of equations. With weak problems it shares the disadvantage of requiring integration, while it has the additional disadvantage of working with higher-order derivatives than weak techniques. It also requires additional regularity in excess of L_2 to conclude that numerical integration of residuals has a controllable error.

For L_∞ residual minimization for problems in strong form, we get a semi-infinite linear programming problem. Application-oriented users should know that there are good numerical techniques for solving such problems. Furthermore, Kuhn–Tucker conditions will help to reduce complexity, as for learning algorithms via support vector machines, while adaptivity on the test side is built in automatically. Thus there is some hope that linear programming codes will be very helpful in the future when it comes to calculate low-complexity solutions of partial differential equations by adaptive methods.

For both cases of residual minimization, there is a trial function with small residuals, if the true solution u has a good approximation $\hat{u}_r$ from the trial space U_r. We avoid h here and prefer r, because trial spaces should not be automatically connected to space discretizations with fill distance h as in finite elements. The existence of $\hat{u}_r$ is a problem of approximation theory which is dependent on the solution u, the trial space U_r, and the norm $\|\cdot\|_U$ in the solution space U only, but not on any partial differential equation. Thus the user should keep Guidelines 3.9, 12.4, and 12.5 in mind without looking at the partial differential equation. Then the numerical method for solving a PDE problem, in weak or strong form, just has to make sure not to discard the existing unknown good approximation $\hat{u}_r$, while it produces another approximation $\tilde{u}_r \in U_r$ based on PDE data which is not too much worse. For residual minimization algorithms, this means that there exists an admissible trial function yielding small residuals $\|L(u) - L(\hat{u}_r)\|_F$, such that the final optimal solution cannot have worse residuals. Error bounds and convergence results will then follow the simple estimates

$$\begin{aligned}\|u - \tilde{u}_r\|_U &\le C\|L(u) - L(\tilde{u}_r)\|_F \\ &= C \inf_{v \in U_r} \|L(u) - L(v)\|_F \\ &\le C\|L(u) - L(\hat{u}_r)\|_F,\end{aligned}$$

which is a well-known line of argument, known in finite elements as Cea's lemma.

Guideline 12.13. Residual minimization works if the problem is well posed and if the trial space contains a good approximation to the solution. This allows plenty of freedom to design useful residual minimization algorithms.

12.8. Discrete residual minimization

Because all residual-based techniques have to evaluate norms on the test side, they have problems when dealing with global L_2, L_∞, or Sobolev norms there. Therefore we now look at *discretization* on the test side. It means that only finitely many 'tests' are performed. Discretization of a strong problem means taking a finite subset of points where the differential equation or boundary conditions are satisfied. This is the standard technique of *collocation.* For weak problems, discrete testing means taking inner products of the residuals with finitely many test functions, and then the residuals are not zero or small, but orthogonal to the *test space* spanned by *test functions.* In both cases we have to make sure that small results of discrete testing lead to small results in (theoretical) infinite testing.

Guideline 12.14. Coping with only finitely many conditions on the test side is the most serious part of any error or convergence analysis for numerical methods solving partial differential equations.

Such an analysis usually requires a *stability* condition relating the test and the trial space, and making sure that a small discrete residual on the trial space implies a small full residual on the trial space. We shall see examples later, but we can already state at this point that there should be no nonzero trial function $\hat{u}_r$ with vanishing test residuals, if we want to have error bounds, because all functions $\tilde{u}_r + \alpha \cdot \hat{u}_r$ for arbitrary $\alpha \in \mathbb{R}$ would have the same discrete test residuals and spoil the error bound.

Guideline 12.15. The discretized residual norm on the test side should at least work like a norm on the trial space.

This is the core of recent work (Schaback 2005*b*) on convergence analysis of unsymmetric methods on which we will now focus.

12.9. Symmetric and unsymmetric methods

Following Guidelines 12.10 and 12.13, the test side will need more attention than the trial side, and this leads us to the distinction between *symmetric* and *unsymmetric* methods. Symmetric methods use discretizations with

- the same degree of freedom on the trial and test side,
- closely related test functionals and trial functions,
- square and possibly positive definite matrices.

For weak problems, this means that trial and test *functions* coincide, and usually the standard Galerkin method is employed, yielding a positive definite square matrix. This applies to finite elements and several generalizations, *e.g.*, the GFEM (Babuška *et al.* 2003), described in detail in this series. The GFEM is a meshless method which enlarges the admissible trial spaces far beyond classical piecewise polynomial finite elements, but still uses the basic symmetric Hilbert space formulation of the finite element method. In its actual form, the GFEM uses stationary scales of trial spaces spanned by a partition of unity. Since it is a symmetric Galerkin technique, the trial functions and the test functions coincide. Compactly supported kernels occur naturally in the partition of unity, but they need not be positive definite. Since the current theory uses stationary approximations (see Section 7) in its scales of local trial spaces, the only kernels providing useful approximation orders are conditionally positive definite with infinite support, like multiquadrics or thin-plate splines. When local trial spaces are generated by moving least squares (see Section 7), weight kernels occur again. But most applications just augment finite element spaces by useful additional trial functions, *e.g.*, for treating singularities. However, the overall axiomatic structure of the GFEM theory (Babuška *et al.* 2003) suggests that it should be possible to extend the theory of the GFEM to allow nonstationary scales of kernel-based trial spaces with high-approximation orders.

For strong problems, the test side contains point evaluation *functionals* and there are no test *functions*. But there is also a symmetric method taking the trial functions as results when these functionals are applied to one argument of a positive definite kernel. This establishes a close relation

$$\lambda \leftrightarrow v_\lambda := \lambda^x K(x,\cdot)$$

between test functionals λ and trial functions v_λ which is only possible because kernels are involved. We call this *symmetric collocation* and deal with it in Section 14. It follows the lines of *general recovery* in Section 3, leading to symmetric positive definite systems of the form (3.8).

Both kinds of symmetric methods can be rewritten as an approximation or optimization problem in Hilbert space, and their theoretical foundation strongly relies on this fact. This comes close to Guideline 3.18, because the problem itself is a quadratic optimization problem solved via a linear system.

Let us now look at unsymmetric methods. In the strong case, collocation (Kansa 1986) using nonstationary scales of trial spaces of radial basis functions, in particular multiquadrics occurs in many applications we cannot list here. Theoretical support was only given recently (Schaback 2005*b*), proving high convergence rates depending on the regularity assumptions. We provide more details in Section 14.

Unsymmetric methods for weak problems usually take the form of Petrov–Galerkin schemes, where trial and test functions differ. Their basic theory (Douglas, Dupont, Rachford and Wheeler 1977) was established for trial spaces spanned by multivariate polynomial splines and for elliptic problems, making use of coercivity. More modern applications (Bialecki and Fairweather 2001, Bialecki, Ganesh and Mustapha 2004) have the same theoretical basis, but also do not apply kernel techniques.

A more radical approach to solving weak problems by an unsymmetric Petrov–Galerkin technique is the *meshless local Petrov–Galerkin* (MLPG) technique developed by S. N. Atluri and his collaborators (Atluri and Zhu 1998) with a short and recent survey (Atluri and Shen 2005) and two books (Atluri and Shen 2002, Atluri 2005) reporting many successful applications. It can use a variety of test and trial functions, and owing to its general form it can claim to include formally many other methods, *e.g.*, Kansa's unsymmetric collocation and various forms of symmetric methods, meshless or not.

However, there is currently no general convergence proof or error estimate available unless the method is restricted to well-known special cases. The main obstacle for its analysis is the fact that it uses a practically very valuable *local weak form* which, as opposed to weak forms arising in standard or generalized finite element methods, cannot be written as a necessary condition for a minimizing trial function in some Hilbert space of functions. But the MLPG can be viewed as an unsymmetric technique which tries to minimize residuals, and thus there are good chances to use Guideline 12.13 for underpinning it, extending techniques (Schaback 2005*b*) which currently only handle the special case of strong testing.

12.10. Numerical integration

Let us finally return to numerical integration questions, and let us look at weak problems first. The integrals for stiffness matrix entries within weak problems usually contain products of test and trial functions or derivatives thereof. To make integration easy and precise, test and trial functions have to be chosen carefully and should be closely related. The standard choice of piecewise polynomial trial and test functions in the finite element method achieves this, since the integrals can be done exactly in the case of polyhedral domains, though one has to keep track of the polyhedra carefully.

The integration of test functions against arbitrary functions is required for the inhomogeneities, but this is an issue of the test side, not of the trial side. Anyway, integrating piecewise polynomials on polyhedra needs some domain triangulation first (the primary *mesh*), and then a careful choice of interpolation nodes (or transformation to standard elements) for the integration (the *integration mesh*). Even 'meshless' methods, if they require integration, may sometimes need an integration mesh and are subject to influences of integration error, if they are applied to weak problems.

Using translates of radial kernels on both the trial and test side of weak problems can be equally efficient as finite elements are, if the integration domains do not interfere with boundaries, because the integrals are univariate radial functions which are either analytically known or can be pretabulated. Certain variations of the MLPG method could take advantage of this. Integration of 'test' kernels against given functions may be simplified by first representing the function in terms of translates of a 'trial' kernel, followed by integrations of kernels against kernels, which again is easy if no boundaries are in the way. In the presence of nontrivial boundaries, all trial and test functions cause problems, unless the real boundary is replaced piecewise by boundaries of supports of trial and test functions.

For problems in strong form, this discussion is not necessary. The trial functions can be chosen freely to satisfy the first three properties of Guideline 12.3. These properties are independent of PDE solving. We shall take a closer look at them, but from a more general point of view.

12.11. Classification of meshless methods

Summarizing, the universe of time-independent meshless methods can be roughly split into four parts by the dichotomies between strong/weak and symmetric/unsymmetric problems.

Strong problems imply collocation techniques as numerical methods, and then there are the symmetric and unsymmetric meshless collocation methods we describe in Section 14. They have in common the use of nonstationary scales of trial functions based on explicit kernels.

Weak problems in unsymmetric form are handled by Petrov–Galerkin techniques or the more general MLPG method. Everything else falls into the category of symmetric techniques solving weak problems. These come in a big variety and mostly differ on the trial side, while one of their common features is to rely on minimization in Hilbert space.

We have to leave out time-dependent meshless methods for space reasons, but we want to point out that there are strange gaps in the above scenario. First, for strong problems there are no investigations of methods using stationary scales. Second, for weak problems there are no investigations of methods using nonstationary scales, though this should be possible using the partition-of-unity framework behind the generalized finite element method. There is plenty of leeway for future research.

13. Special meshless kernel techniques

Following Guideline 6.1 and applying techniques of Section 6, kernel engineering can provide kernels which are closely connected to standard differential equations. This is used by certain numerical methods to be described in this section.

13.1. Dual reciprocity method

This misleading name stands for a technique coming from boundary element methods (Nardini and Brebbia 1982, Partridge, Brebbia and Wrobel 1992) and proliferating by use of kernel techniques (Chen, Golberg and Schaback 2003*a*). The basic idea is to split the problem into an inhomogeneous and a homogeneous subproblem with respect to the differential equation. A problem $L(u) = f$ with a linear differential operator L and linear boundary conditions $B(u) = g$ is treated first by constructing a *particular* solution u_P with $L(u_P) = f$ without regard of boundary values. Then the *homogeneous* problem $L(u) = 0$ is solved by some function u_H under the boundary conditions $B(u) = g - B(u_P)$ to get the final solution as $u := u_P + u_H$.

The first problem uses trial spaces of known particular solutions. These are easy to construct for kernel-based trial functions. The second problem makes use of *a priori* information on homogeneous solutions either via *integral equations* or *fundamental solutions*, providing trial spaces of homogeneous solutions via a special kernel called *the* fundamental solution of the differential operator L. Because of this close connection to kernels, we have to treat this technique in some detail.

Guideline 13.1. The dual reciprocity method can be applied to well-posed linear problems with well-known fundamental and particular solutions which have good approximation properties.

13.2. Method of particular solutions

To find a *particular* solution u_P with $L(u_P) = f$ without regard to boundary values, one can use trial functions u_i whose images $f_i := L(u_i)$ under L are well known and numerically available. Then the right-hand side f of the differential equation is approximated by a linear combination

$$\tilde{f} := \sum_i \alpha_i f_i$$

of the f_i to some small error $\|f - \tilde{f}\|_F$ in some suitable function space F, and the approximation

$$\tilde{u}_P := \sum_i \alpha_i u_i$$

is the canonical approximation to a particular solution u_P. Note that this part of the algorithm is an approximation problem which is completely independent of partial differential equations. After construction of $\tilde{f}$ we know that the *residual* $L(u_P - \tilde{u}_P) = f - \tilde{f}$ is small, but we have to postpone a thorough error analysis based on residual minimization and Guideline 12.13 until we have looked at the homogeneous problem and boundary conditions.

Of course, there are plenty of ways to produce good approximations $\tilde{f}$ to f, provided that the approximation properties of the functions f_i are well known. But it is a problem to find functions f_i which are particular solutions *and* have good approximation properties. Starting from well-approximating multivariate functions f_i such as finite elements, it is often hard or impossible to find the functions u_i with $L(u_i) = f_i$. On the other hand, starting with nice functions u_i will only rarely lead to functions $f_i = L(u_i)$ with good approximation properties.

But things can be easy if kernels are used. The simplest way is to take a smooth symmetric translation-invariant positive definite kernel K and define

$$u_i := K(\cdot - x_i) \quad \text{and} \quad f_i := LK(\cdot - x_i)$$

for trial centres x_i. If the operator L is elliptic with constant coefficients, the resulting kernel LK for the f_i will be positive definite again, as inspection of Fourier transforms shows. Now all techniques of Section 7 can be applied to reconstruct f approximately using the trial functions f_i.

If the operator is not elliptic, the kernel LK will not be positive definite. In such cases, the reverse strategy can be helpful, starting with $f_i := K(\cdot - x_i)$ using a positive definite kernel K and finding another kernel K_L such that $L(K_L) = K$. This new kernel need not be positive definite, but since it is not used for approximation, there is no problem here.

Guideline 13.2. A natural kernel-based strategy for the method of particular solutions is to have pairs $u_i,\ f_i$ with $f_i = L(u_i) = K(\cdot - x_i)$ such that one can perform approximation of f by the standard translates of the kernel K.

The literature contains many such pairs, and we can only cite a selection: Chen and Rashed (1998), Chen, Muleshkov and Golberg (1999*b*), Cheng (2000), Ramachandran and Balakrishnan (2000) and Golberg, Muleshkov, Chen and Cheng (2003).

13.3. Method of fundamental solutions

Once the problem $L(u) = f$ with boundary data $B(u) = g$ is transformed into homogeneous form $L(u) = 0,\ B(u) = g - B(\tilde{u}_P) =: g_H$ by the method of *particular solutions*, the method of *fundamental solutions* (Mathon and Johnston 1977, Fairweather and Karageorghis 1998) takes over. It uses a special kernel F called *the* fundamental solution of $L(u) = 0$ such that $LF(\cdot, x) = \delta_x$ in the distributional sense. These kernels are well known for a number of linear operators, and we presented those for the iterated Laplacian in Section 6, *i.e.*, the *thin-plate spline* of (6.1) and the *polyharmonic splines* of (6.2). This can be generalized to linear elliptic differential operators with constant coefficients, but we do not want to go into details and refer the reader to the literature on Fourier methods in partial differential equations (Hörmander 2003) and on special fundamental solutions (Kythe 1996, Golberg and Chen 1999, Chen, Marcozzi and Choi 1999*a*, Balakrishnan and Ramachandran 2000, Alves, Chen and Šarler 2002, Poullikkas, Karageorghis and Georgiou 2002, Hon and Wei 2004), where we again picked out just a few cases from different application areas.

However, as we pointed out at the end of Section 6, the kernel F providing the fundamental solution will have a singularity 'on the diagonal', *i.e.*, for $F(x, x)$ or derivatives thereof. For second-order equations in dimension 2 or more, F itself is already singular, while for higher order we get singularities in the derivatives of F. Singular kernels are not directly covered by the standard theory of positive definite kernels, but they work fine in the generalized sense of (6.5), avoiding point evaluation functionals.

Once a fundamental solution F is at hand, there are various ways to generate trial functions solving the homogeneous differential equation. Before we describe these techniques, we want to look at the error and convergence analysis. The trial functions are used for approximating the prescribed boundary values g_H on the boundary. If a numerical scheme comes up with a trial function $\tilde{u}_H$ satisfying $L(\tilde{u}_H) = 0$ and with a small residual $B(\tilde{u}_H) - g_H = B(\tilde{u}_H) - B(u) + B(\tilde{u}_P)$, we use $\tilde{u} := \tilde{u}_H + \tilde{u}_P$ for our full solution and residuals

$$\begin{aligned}
\|L(\tilde{u}) - f\|_F &= \|L(\tilde{u}_H + \tilde{u}_P) - f\|_F \\
&= \|L(\tilde{u}_P) - f\|_F \\
&= \|\tilde{f} - f\|_F, \\
\|B(\tilde{u}) - g\|_G &= \|B(\tilde{u}_H + \tilde{u}_P) - g\|_G \\
&= \|B(\tilde{u}_H) - g_H\|_G,
\end{aligned}$$

for a suitable norm on a space G where the boundary values live. If the problem is continuously dependent on the data in the sense that an *a priori* inequality

$$\|u\|_U \leq C(\|L(u)\|_F + \|B(u)\|_G) \tag{13.1}$$

holds, and if the exact solution u exists and lies in U, then there is an error bound

$$\begin{aligned}\|\tilde{u} - u\|_U &\leq C(\|L(\tilde{u} - u)\|_F + \|B(\tilde{u} - u)\|_G)\\ &= C(\|L(\tilde{u}) - f\|_F + \|B(\tilde{u}) - g\|_G)\\ &= C(\|\tilde{f} - f\|_F + \|B(\tilde{u}_H) - g_H\|_G),\end{aligned}$$

reducing the overall error to the error of the residuals.

Guideline 13.3. The dual reciprocity method has a solid mathematical foundation once continuous dependence holds and the residuals are small in the correct norms.

This confirms Guidelines 13.2 and 12.13. For elliptic operators satisfying a maximum principle, these error bounds can be improved, provided that the spaces F and G are chosen appropriately.

However, it remains to prove that certain approximation schemes in the methods of particular and fundamental solutions lead to small residuals in the correct spaces needed for continuous dependence. If methods of Section 7 based on positive definite kernels are applied within the method of particular solutions, there are no serious problems, because there are good error estimates like (7.12) in Sobolev spaces on bounded domains. We are thus left with the analysis of the approximation power of the method of fundamental solutions.

A particularly simple way to generate trial functions satisfying the homogeneous problem $L(u) = 0$ is to proceed as in Section 7 by taking linear combinations of translates $F(\cdot, x_i)$ of the fundamental solution. This is an approximation problem

$$g_H(t) \approx \sum_j \alpha_j F(t, x_j), \quad t \in \Gamma \tag{13.2}$$

to be posed on the boundary Γ of the domain. But in order to avoid singularities of trial functions inside the domain or on the boundary, the trial centres x_i should be placed outside the domain. But then the theory of Section 7 does not apply, because the approximation domain does not contain the centres and there is no notion like a fill distance as in (7.6) making sense. However, the references cited above support that this method performs very well in practice if the outside centres are placed with care. For very special domains and smooth boundary data the method can be proven to have spectral convergence (Li 2005), but a general theory is still missing.

A well-known and much older approach is to place infinitely many trial centres right on the boundary and to take a weighted sum over all such translates of the fundamental solution. This leads to the singular *single-layer potential* integral equation

$$g_H(t) = \int_\Gamma \alpha(x) F(t,x)\,\mathrm{d}x, \quad t \in \Gamma.$$

Note that this is a non-discrete form of (13.2). Owing to the singularities of F, this equation cannot be solved strongly, but it can be solved weakly, *e.g.*, via finite elements on the boundary. Such techniques are called *boundary element methods* and have a rich literature including various books.

Variations of this approach are possible by replacing $F(\cdot, x)$ by certain linear functionals acting on $F(\cdot, x)$ with respect to the second argument x. These new kernels, like the normal derivative $\frac{\partial F(\cdot,x)}{\partial n}$ will usually preserve the property that action of L on the first argument results in zero. The standard case is the integral equation of the *double-layer potential*, but there are plenty of other possibilities that are yet unexploited, *e.g.*, replacing $F(t,x)$ by local integrals around x of $F(t,s)$ with respect to s in order to remove the singularities. A special case of this is the recent *boundary knot method* (Chen and Tanaka 2002, Chen 2002, Chen and Hon 2003).

13.4. Divergence-free kernels

In analogy to the methods of fundamental and particular solutions, there is a trick (Narcowich and Ward 1994*a*, Lowitzsch 2005) to generate kernel-based divergence-free trial spaces from smooth kernels, and curl-free trial spaces are also possible. The general idea behind this is to employ *matrix-valued* kernels, which allow us to incorporate these additional features into the rows and/or columns of the matrix. From these matrix-valued kernels, *vector-valued* interpolants can be built, which satisfy the additional constraints analytically. Applications of such divergence-free kernels to Stokes, Navier–Stokes, Euler and Maxwell equations are currently under investigation. We see this as a further case of *kernel engineering* in the direction of partial differential equations.

Unfortunately, we cannot describe more general applications of kernels to transport problems, advection, and fluid dynamics here, but this is a very promising research area (Mai-Duy and Tran-Cong 2001, Behrens and Iske 2002, Iske 2003, Barba, Leonard and Allen 2005, Shu, Ding and Yeo 2005).

14. Meshless collocation

Within the classification of meshless methods in Section 12, the techniques of this section solve partial differential equations in *strong* form, using *collocation* on the test side and avoiding *numerical integration* completely. On the trial side, they use *nonstationary* scales of *explicit* kernel-based trial functions. They come in a *symmetric* and an *unsymmetric* form.

In both cases, a given partial differential equation $L(u) = f$ and various boundary conditions of the form $B(u) = g$ are discretized by point evaluations of both sides in certain *collocation* nodes. For instance, a Poisson problem on a domain Ω with Dirichlet conditions $u = g_D$ on $\Gamma_D \subseteq \Gamma := \partial\Omega$ and Neumann conditions $\frac{\partial u}{\partial n} = g_N$ on $\Gamma_N \subset \Gamma$ can be discretized by a set $\Lambda := \{\lambda_1, \dots, \lambda_N\}$ of *test functionals* consisting of three parts:

$$\begin{aligned}
\Lambda &= \Lambda_1 \cup \Lambda_2 \cup \Lambda_3 \\
\Lambda_1 &:= \{\lambda_1, \dots, \lambda_{N_1}\} \\
\lambda_j(u) &:= -\Delta u(x_j), && x_j \in \overline{\Omega},\ 1 \le j \le N_1, \\
\Lambda_2 &:= \{\lambda_{1+N_1}, \dots, \lambda_{N_2}\} \\
\lambda_j(u) &:= u(x_j), && x_j \in \Gamma_D,\ 1 + N_1 \le j \le N_2, \\
\Lambda_3 &:= \{\lambda_{1+N_2}, \dots, \lambda_{N_3}\} \\
\lambda_j(u) &:= \tfrac{\partial u}{\partial n}(x_j), && x_j \in \Gamma_N,\ 1 + N_2 \le j \le N_3 =: N.
\end{aligned}$$

If the evaluation points within the three sets $\Lambda_1,\ \Lambda_2,\ \Lambda_3$ of functionals are different, all linear test functionals in $\Lambda = \Lambda_1 \cup \Lambda_2 \cup \Lambda_3$ are linearly independent.

This specifies the *test* part of the problem for both the symmetric and unsymmetric methods. In general, there may be several differential operators and several boundary conditions in any kind of mixture, provided that everything is linear in u and the test functionals are linearly independent. There is no numerical integration, no test functions, and up to now there are no kernels. But for practical reasons, we mention the following guideline.

Guideline 14.1. To give certain test functionals special importance, one should apply constant factors.

For example, boundary test functionals in two-dimensional Poisson problems should get a factor of about 1000 over the differential equation test functionals. Exact rules for this are not known, but the background is provided by continuous dependence inequalities such as (13.1) where the parts of the right-hand side should carry different weights.

14.1. Symmetric meshless collocation

The difference between symmetric and unsymmetric meshless collocation shows up when looking at the trial side, provided that they use the same testing strategy. For *unsymmetric* collocation, a standard nonstationary scale of kernel-based trial spaces is used, where the translates $K(\cdot, y_k)$ are taken with *trial* nodes y_k that are independent of the *test functionals*. This method goes back to Kansa (1986) and will be analysed later.

In the symmetric case, there must be a strong connection between trial functions and test functionals. This is done by taking the trial functions $\lambda_j^x K(\cdot, x)$, $1 \leq j \leq N$ for a sufficiently smooth kernel K guaranteeing that all test functionals $\lambda_1, \ldots, \lambda_N$ lie in the dual of its native space. This is a special case of general Hermite–Birkhoff interpolation as described in Section 3. Under mild additional assumptions, this leads to a symmetric nonsingular linear system (3.8) and error bounds along the lines of Section 7. A detailed theoretical analysis of symmetric collocation can be found in the literature (Wu 1992, Iske 1995, Franke and Schaback 1998*a*, 1998*b*), while reports on applications are somewhat scattered (Power and Barraco 2002, Larsson and Fornberg 2003, Fasshauer 2004, Šarler 2005) and often limited to small problems with regular solutions. For such cases, the method gives quick and useful results, provided that the general guidelines on scaling in Section 3 are observed. Future work should apply special techniques of Section 8 for handling large-scale and ill-conditioned systems.

14.2. Unsymmetric meshless collocation

Unsymmetric meshless collocation is much more popular than the symmetric case, because it is easier to handle and shows similarly good experimental results (Cheng, Golberg, Kansa and Zammito 2003). The matrix entries $\lambda_i^x \lambda_j^y K(x, y)$ of the symmetric case apply all derivatives twice, while the unsymmetric case with trial functions $K(\cdot, y_k)$ involves only $\lambda_i^x K(x, y_k)$, which is simpler to program. There is a huge number of papers on practical applications of this technique which we cannot cite here, unfortunately, and which would require a survey of its own. Some application areas with recent sample papers are

- convection-diffusion problems (Li and Chen 2003, La Rocca, Hernandez Rosales and Power 2005),
- ill-posed problems (Cheng and Cabral 2005),
- thermal analysis (Pepper and Šarler 2005),
- fluid dynamics (Šarler 2005),
- flows in porous media (Šarler, Perko and Chen 2004),
- viscous vortex flows (Barba *et al.* 2005),
- boundary-layer problems (Ling and Trummer 2004),
- transport problems (Lorentz, Narcowich and Ward 2003),
- free boundary value problems (Kovačevič, Poredoš and Šarler 2003),
- fracture problems (Lee and Yoon 2004),
- nonlinear problems in smart materials (Liu, Liew, Hon and Zhang 2005),

but this list is far from complete. These papers, however, are recent enough to enable readers starting in these areas to find older results and the application-oriented background.

A thorough theoretical analysis was missing for about 20 years, because the unsymmetric systems can in general be singular (Hon and Schaback 2001). However, if the method is changed along some of the guidelines of this survey, error bounds and a convergence analysis can be supplied (Schaback 2005*b*). We summarize the relevant issues as follows.

Guideline 14.2. The mathematical foundation of unsymmetric collocation requires four ingredients:

(a) a linear and well-posed PDE problem,

(b) a nonstationary scale of meshless trial spaces with good approximation properties and spanned by sufficiently smooth kernel translates $K(\cdot, y_k)$,

(c) a scale of test discretizations via sets of collocation functionals λ_i which is fine enough to guarantee at least a full rank of the unsymmetric linear systems with entries $\lambda_i^x K(x, y_k)$,

(d) an approximate solution of this linear system with small discrete residuals.

Items (a)–(c) above are (in a more detailed and rigid form) sufficient to guarantee approximate solvability in the final step. It can be implemented by various techniques including linear or least-squares optimization or greedy adaptive methods (Hon *et al.* 2003, Ling and Schaback 2004) described below. Of course, guidelines of Section 7 concerning scaling must be observed at all times. If the sup norm of residuals is minimized, the method reduces to linear optimization, and it can be implemented via the revised simplex method. By the Kuhn–Tucker theory, the final result will then be based only on a small finite set of test functionals. This is a connection to *support vector machines.*

14.3. Adaptive collocation solvers

In finite elements, there is a vast recent literature on adaptivity controlled by efficient error estimation techniques. Meshless kernel-based collocation methods can implement this in a very simple way by inspecting residuals of the differential equation and the boundary conditions. Since evaluations of trial functions are explicitly possible and very cheap, one can always evaluate the residuals on a large set of background test points, using only a few of these to define the test functionals entering the calculations.

The general recipe is as follows.

1 Start with $\tilde{u}$ being the zero trial function and set $N := 0$.

2 Iteration:
Assume that there is a trial function $\tilde{u}$ which is a linear combination of N trial functions $u_1, \ldots, u_N$ such that the $N \times N$ system with entries $\lambda_j(u_k)$ for N test functionals $\lambda_1, \ldots, \lambda_N$ is non-singular and has $\tilde{u}$ as an approximate solution.

   (a) Find a point in the domain or on the boundary where there is a large or maximal residual. Stop if none can be found.

   (b) Use this point to define a new test functional λ_{N+1} for further calculations.

   (c) Add a new trial function u_{N+1} such that the enlarged system still is nonsingular.

   (d) Solve the new system approximately for a new trial function $\tilde{u}$.

If candidates for test functionals and trial functions are chosen from a large reservoir satisfying the background theory for unsymmetric calculations as described in Guideline 14.2, this is an adaptive bootstrapping technique that automatically selects useful subsets of trial functions and test functionals without ever forming a huge matrix defined by all possible trial functions and test functionals. Connections to the notions of *dictionaries* in approximation theory and to *greedy algorithms* are apparent.

This technique works fine for small problems (Hon *et al.* 2003, Ling and Schaback 2004) but needs further theoretical and numerical research if N gets large and the systems get ill-conditioned. In particular, step 2(c) of the algorithm can be implemented in various ways, and it is not clear how to assess the performance for small N. The symmetric case can also be handled adaptively by omitting step 2(c), taking the new test point and the corresponding functional to define a new trial function. There is a theoretical background for this symmetric *greedy* strategy in the interpolation case (Schaback and Wendland 2000*a*), but a thorough analysis for general collocation is an open problem, as is the incorporation of methods from Section 8 dealing with large ill-conditioned systems.

REFERENCES

R. Ahrem, A. Beckert and H. Wendland (2005), A new multivariate interpolation method for large-scale spatial coupling problems in aeroelasticity. DGLR-Bericht 2005-04; *Proc. IFASD*, Munich 2005.

C. J. S. Alves, C. S. Chen and B. Šarler (2002), The method of fundamental solutions for solving Poisson problems, in *Boundary Elements XXIV, Sintra 2002*, Vol. 13 of *Int. Ser. Adv. Bound. Elem.*, WIT Press, Southampton, pp. 67–76.

A. Appel (1985), 'An efficient program for many-body simulation', *SIAM J. Sci. Statist. Comput.* **8**, 85.

N. Aronszajn (1950), 'Theory of reproducing kernels', *Trans. Amer. Math. Soc.* **68**, 337–404.

S. N. Atluri (2005), *The Meshless Method (MLPG) for Domain and BIE Discretizations*, Tech Science Press, Encino, CA.

S. N. Atluri and S. Shen (2002), *The Meshless Local Petrov–Galerkin (MLPG) Method*, Tech Science Press, Encino, CA.

S. N. Atluri and S. Shen (2005), 'The basis of meshless domain discretization: the meshless local Petrov–Galerkin (MLPG) method', *Adv. Comput. Math.* **23**, 73–93.

S. N. Atluri and T. L. Zhu (1998), 'A new meshless local Petrov–Galerkin (MLPG) approach to nonlinear problems in computer modeling and simulation', *Computer Modeling and Simulation in Engineering* **3**, 187–196.

M. Atteia (1992), *Hilbertian Kernels and Spline Functions*, North-Holland, Amsterdam, Vol. 4 of *Studies in Computational Mathematics*.

I. Babuška and J. M. Melenk (1997), 'The partition of unity method', *Internat. J. Numer. Methods Engrg.* **40**, 727–758.

I. Babuška, U. Banerjee and J. E. Osborn (2002), Meshless and generalized finite element methods: A survey of some major results, in *Meshfree Methods for Partial Differential Equations*, Vol. 26 of *Lecture Notes in Computational Science and Engineering*, Springer, pp. 1–20.

I. Babuška, U. Banerjee and J. E. Osborn (2003), Survey of meshless and generalized finite element methods: A unified approach, in *Acta Numerica*, Vol. 12, Cambridge University Press, pp. 1–215.

K. Balakrishnan and P. A. Ramachandran (2000), 'The method of fundamental solutions for linear diffusion-reaction equations', *Math. Comput. Modelling* **31**, 221–237.

K. Ball (1992), 'Eigenvalues of Euclidean distance matrices', *J. Approx. Theory* **68**, 74–82.

K. Ball, N. Sivakumar and J. D. Ward (1992), 'On the sensitivity of radial basis interpolation to minimal data separation distance', *Constr. Approx.* **8**, 401–426.

L. A. Barba, A. Leonard and C. B. Allen (2005), 'Advances in viscous vortex methods: Meshless spatial adaption based on radial basis function interpolation', *Internat. J. Numer. Meth. Fluids* **47**, 387–421.

J. E. Barnes and P. Hut (1986), 'A hierarchical $\mathcal{O}(N \log N)$ force-calculation algorithm', *Nature* **324**, 446–449.

R. K. Beatson and E. Chacko (2000), Fast evaluation of radial basis functions: A multivariate momentary evaluation scheme, in *Curve and Surface Fitting: Saint-Malo 1999* (A. Cohen, C. Rabut and L. L. Schumaker, eds), Vanderbilt University Press, Nashville, pp. 37–46.

R. K. Beatson and L. Greengard (1997), A short course on fast multipole methods, in *Wavelets, Multilevel Methods and Elliptic PDEs; 7th EPSRC Numerical Analysis Summer School, University of Leicester, Leicester, GB, July 8–19, 1996* (M. Ainsworth, J. Levesley, W. Light and M. Marletta, eds), Clarendon Press, Oxford, pp. 1–37.

R. K. Beatson and W. A. Light (1993), ‘Quasi-interpolation with thin plate splines on a square’, *Constr. Approx.* **9**, 407–433.

R. K. Beatson and W. A. Light (1997), ‘Fast evaluation of radial basis functions: Methods for two-dimensional polyharmonic splines’, *IMA J. Numer. Anal.* **17**, 343–372.

R. K. Beatson and G. N. Newsam (1992), ‘Fast evaluation of radial basis functions: I’, *Comput. Math. Appl.* **24**, 7–19.

R. K. Beatson and G. N. Newsam (1998), ‘Fast evaluation of radial basis functions: Moment-based methods’, *SIAM J. Sci. Comput.* **19**, 1428–1449.

R. K. Beatson, J. B. Cherrie and C. T. Mouat (1999), ‘Fast fitting of radial basis functions: Methods based on preconditioned GMRES iteration’, *Adv. Comput. Math.* **11**, 253–270.

R. K. Beatson, J. B. Cherrie and D. L. Ragozin (2000*a*), Polyharmonic splines in $\mathbb{R}^d$: Tools for fast evaluation, in *Curve and Surface Fitting: Saint-Malo 1999* (A. Cohen, C. Rabut and L. L. Schumaker, eds), Vanderbilt University Press, Nashville, pp. 47–56.

R. K. Beatson, J. B. Cherrie and D. L. Ragozin (2001), ‘Fast evaluation of radial basis functions: Methods for four-dimensional polyharmonic splines’, *SIAM J. Math. Anal.* **32**, 1272–1310.

R. K. Beatson, G. Goodsell and M. J. D. Powell (1996), ‘On multigrid techniques for thin plate spline interpolation in two dimensions’, *Lect. Appl. Math.* **32**, 77–97.

R. K. Beatson, W. A. Light and S. Billings (2000*b*), ‘Fast solution of the radial basis function interpolation equations: Domain decomposition methods’, *SIAM J. Sci. Comput.* **22**, 1717–1740.

A. Beckert (2000), ‘Coupling fluid (CFD) and structural (FE) models using finite interpolation elements’, *Aerosp. Sci. Technol.* **1**, 13–22.

A. Beckert and H. Wendland (2001), ‘Multivariate interpolation for fluid-structure-interaction problems using radial basis functions’, *Aerosp. Sci. Technol.* **5**, 125–134.

J. Behrens and A. Iske (2002), ‘Grid-free adaptive semi-Lagrangian advection using radial basis functions’, *Comput. Math. Appl.* **43**, 319–327.

T. Belytschko, Y. Krongauz, M. Fleming, D. Organ and W. K. Liu (1996*a*), ‘Smoothing and accelerated computations in the element free Galerkin method’, *J. Comput. Appl. Math.* **74**, 111–126.

T. Belytschko, Y. Krongauz, D. Organ, M. Fleming and P. Krysl (1996*b*), ‘Meshless methods: An overview and recent developments’, *Comput. Methods Appl. Mech. Engrg.* **139**, 3–47.

B. Bialecki and G. Fairweather (2001), ‘Orthogonal spline collocation methods for partial differential equations’, *J. Comput. Appl. Math.* **128**, 55–82.

B. Bialecki, M. Ganesh and K. Mustapha (2004), ‘A Petrov–Galerkin method with quadrature for elliptic boundary value problems’, *IMA J. Numer. Anal.* **24**, 157–177.

P. Binev and K. Jetter (1992), Estimating the condition number for multivariate interpolation problems, in *Numerical Methods in Approximation Theory, Vol. 9:*

Proc. Conf. Oberwolfach, Germany, November 24–30, 1991 (D. Braess *et al.*, eds), Vol. 105 of *International Series of Numerical Mathematics*, Birkhäuser, Basel, pp. 41–52.

M. Bozzini, L. Lenarduzzi and R. Schaback (2002), 'Adaptive interpolation by scaled multiquadrics', *Adv. Comput. Math.* **16**, 375–387.

D. Brown, L. Ling, E. Kansa and J. Levesley (2005), 'On approximate cardinal preconditioning methods for solving PDEs with radial basis functions', *Engrg. Anal. Boundary Elements* **19**, 343–353.

M. D. Buhmann (1988), 'Convergence of univariate quasi-interpolation using multiquadrics', *IMA J. Numer. Anal.* **8**, 365–383.

M. D. Buhmann (1990), 'Multivariate cardinal interpolation with radial-basis functions', *Constr. Approx.* **6**, 225–255.

M. D. Buhmann (1993), 'On quasi-interpolation with radial basis functions', *J. Approx. Theory* **72**, 103–130.

M. D. Buhmann (1998), 'Radial functions on compact support', *Proc. Edinb. Math. Soc. II* **41**, 33–46.

M. D. Buhmann (2000), 'Radial basis functions', in *Acta Numerica*, Vol. 9, Cambridge University Press, pp. 1–38.

M. D. Buhmann (2004), *Radial Basis Functions*, Cambridge Monographs on Applied and Computational Mathematics, Cambridge University Press.

M. D. Buhmann, N. Dyn and D. Levin (1995), 'On quasi-interpolation by radial basis functions with scattered centers', *Constr. Approx.* **11**, 239–254.

J. C. Carr, R. K. Beatson, J. B. Cherrie, T. J. Mitchell, W. R. Fright, B. C. McCallum and T. R. Evans (2001), Reconstruction and representation of 3D objects with radial basis functions, in *SIGGRAPH '01: Proc. 28th Annual Conf. Computer Graphics and Interactive Techniques*, ACM Press, New York, NY, USA, pp. 67–76.

W. zu Castell and F. Filbir (2005), 'Radial basis functions and corresponding zonal series expansion on the sphere', *J. Approx. Theory* **134**, 65–79.

T. Cecil, J. Qian and S. Osher (2004), 'Numerical methods for high dimensional Hamilton–Jacobi equations using radial basis functions', *J. Comput. Phys.* **196**, 327–347.

C. S. Chen and Y. F. Rashed (1998), 'Evaluation of thin plate spline based particular solutions for Helmholtz-type operators for the DRM', *Mech. Res. Comm.* **25**, 195–201.

C. S. Chen, M. A. Golberg and R. Schaback (2003*a*), Recent developments in the dual reciprocity method using compactly supported radial basis functions, in *Transformation of Domain Effects to the Boundary* (Y. F. Rashed and C. A. Brebbia, eds), WIT Press, Southampton, Boston, pp. 138–225.

C. S. Chen, M. D. Marcozzi and S. Choi (1999*a*), The method of fundamental solutions and compactly supported radial basis functions: A meshless approach to 3D problems, in *Boundary Elements XXI: Oxford, 1999*, Vol. 6 of *Internat. Ser. Adv. Bound. Elem.*, WIT Press, Southampton, pp. 561–570.

C. S. Chen, A. S. Muleshkov and M. A. Golberg (1999*b*), The numerical evaluation of particular solution for Poisson's equation: A revisit, in *Boundary Elements XXI* (C. Brebbia and H. Power, eds), WIT Press, pp. 313–322.

D. Chen, V. A. Menegatto and X. Sun (2003*b*), 'A necessary and sufficient condition for strictly positive definite functions on spheres', *Proc. Amer. Math. Soc.* **131**, 2733–2740.

W. Chen (2002), 'Symmetric boundary knot method', *Engrg. Anal. Boundary Elements* **26**, 489–494.

W. Chen and Y. C. Hon (2003), 'Numerical convergence of boundary knot method in the analysis of Helmholtz, modified Helmholtz, and convection-diffusion problems', *Comput. Methods Appl. Mech. Engrg.* **192**, 1859–1875.

W. Chen and M. Tanaka (2002), 'A meshless, integration-free, and boundary-only RBF technique', *Comput. Math. Appl.* **43**, 379–391.

E. W. Cheney and W. A. Light (2000), *A Course in Approximation Theory*, Brooks/Cole Publishing Company, Pacific Grove.

E. W. Cheney, W. A. Light and Y. Xu (1992), 'On kernels and approximation orders', *Lect. Notes Pure Appl. Math.* **138**, 227–242.

A. H.-D. Cheng (2000), 'Particular solutions of Laplacian, Helmholtz-type, and polyharmonic operators involving higher order radial basis functions', *Engrg. Anal. Boundary Elements* **24**, 531–538.

A. H.-D. Cheng and J. J. S. P. Cabral (2005), Direct solution of certain ill-posed boundary value problems by collocation method, in *Boundary Elements XXVII* (A. Kassab, C. A. Brebbia, E. Divo and D. Poljak, eds), pp. 35–44.

A. H.-D. Cheng, M. A. Golberg, E. J. Kansa and G. Zammito (2003), 'Exponential convergence and *h*-*c* multiquadric collocation method for partial differential equations', *Numer. Methods Partial Differential Equations* **19**, 571–594.

J. Cheng, Y. C. Hon and M. Yamamoto (2001*a*), 'Conditional stability estimation for an inverse boundary problem with non-smooth boundary in $\mathbb{R}^3$', *Trans. Amer. Math. Soc.* **353**, 4123–4138.

J. Cheng, Y. C. Hon, T. Wei and M. Yamamoto (2001*b*), 'Numerical computation of a Cauchy problem for Laplace's equation', *ZAMM Z. Angew. Math. Mech.* **81**, 665–674.

P. G. Ciarlet (1991), Basic error estimates for elliptic problems, in *Handbook of Numerical Analysis*, Vol. II, North-Holland, Amsterdam, pp. 17–351.

D. D. Cox (1984), 'Multivariate smoothing spline functions', *SIAM J. Numer. Anal.* **21**, 789–813.

P. Craven and G. Wahba (1979), 'Smoothing noisy data with spline functions: estimating the correct degree of smoothing by the method of generalized cross-validation', *Numer. Math.* **31**, 377–403.

N. Cristianini and J. Shawe-Taylor (2000), *An Introduction to Support Vector Machines and Other Kernel-Based Learning Methods*, Cambridge University Press, Cambridge.

F. Cucker and S. Smale (2001), 'On the mathematical foundation of learning', *Bull. Amer. Math. Soc.* **39**, 1–49.

N. Dodgson, M. Floater and M. Sabin (2004), *Advances in Multiresolution for Geometric Modelling*, Springer, Berlin, Germany.

J. Douglas, Jr., T. Dupont, H. H. Rachford, Jr. and M. F. Wheeler (1977), 'Local H^{-1} Galerkin procedures for elliptic equations', *RAIRO Anal. Numér.* **11**, 3–12.

T. A. Driscoll and B. Fornberg (2002), 'Interpolation in the limit of increasingly flat radial basis functions', *Comput. Math. Appl.* **43**, 413–422.

M. R. Dubal (1994), 'Domain decomposition and local refinement for multiquadric approximations I: Second-order equations in one-dimension', *J. Appl. Sci. Comput.* **1**, 146–171.

J. Duchon (1976), 'Interpolation des fonctions de deux variables suivant le principe de la flexion des plaques minces', *Rev. Française Automat. Informat. Rech. Opér. Anal. Numer.* **10**, 5–12.

J. Duchon (1979), Splines minimizing rotation-invariate semi-norms in Sobolev spaces, in *Constructive Theory of Functions of Several Variables* (W. Schempp and K. Zeller, eds), Springer, Berlin/Heidelberg, pp. 85–100.

N. Dyn, F. J. Narcowich and J. D. Ward (1999), 'Variational principles and Sobolev-type estimates for generalized interpolation on a Riemannian manifold', *Constr. Approx.* **15**, 174–208.

T. Evgeniou, M. Pontil and T. Poggio (2000), 'Regularization networks and support vector machines', *Adv. Comput. Math.* **13**, 1–50.

G. Fairweather and A. Karageorghis (1998), 'The method of fundamental solution for elliptic boundary value problems', *Adv. Comput. Math.* **9**, 69–95.

C. Farhat and M. Lesoinne (1998), Higher-order staggered and subiteration free algorithms for coupled dynamic aeroelasticity problems, in *36th Aerospace Sciences Meeting and Exhibit, AIAA 98-0516, Reno/NV.*

R. Farwig (1986), 'Multivariate interpolation of arbitrarily spaced data by moving least squares methods', *J. Comput. Appl. Math.* **16**, 79–83.

R. Farwig (1987), Multivariate interpolation of scattered data by moving least squares methods, in *Algorithms for Approximation* (J. C. Mason and M. G. Cox, eds), Clarendon Press, Oxford, pp. 193–211.

R. Farwig (1991), Rate of convergence of moving least squares interpolation methods: The univariate case, in *Progress in Approximation Theory* (P. Nevai and A. Pinkus, eds), Academic Press, Boston, pp. 313–327.

G. Fasshauer (2004), RBF collocation methods and pseudospectral methods. Preprint, `http://amadeus.csam.iit.edu/~fass/`.

G. Fasshauer and L. L. Schumaker (1998), Scattered data fitting on the sphere, in *Mathematical Methods for Curves and Surfaces II* (M. Dæhlen, T. Lyche and L. L. Schumaker, eds), Vanderbilt University Press, Nashville, pp. 117–166.

A. C. Faul and M. J. D. Powell (1999), 'Proof of convergence of an iterative technique for thin plate spline interpolation in two dimensions', *Adv. Comput. Math.* **11**, 183–192.

S. Fleishman, D. Cohen-Or and C. T. Silva (2005), 'Robust moving least-squares fitting with sharp features', *ACM Trans. Graph.* **24**, 544–552.

M. S. Floater and A. Iske (1996), 'Multistep scattered data interpolation using compactly supported radial basis functions', *J. Comput. Appl. Math.* **73**, 65–78.

M. S. Floater and A. Iske (1998), 'Thinning algorithms for scattered data interpolation', *BIT* **38**, 705–720.

B. Fornberg and D. M. Sloan (1994), A review of pseudospectral methods for solving partial differential equations, in *Acta Numerica*, Vol. 3, Cambridge University Press, pp. 203–267.

C. Franke and R. Schaback (1998*a*), 'Convergence order estimates of meshless collocation methods using radial basis functions', *Adv. Comput. Math.* **4**, 381–399.

C. Franke and R. Schaback (1998*b*), 'Solving partial differential equations by collocation using radial basis functions', *Appl. Math. Comput.* **93**, 73–82.

W. Freeden, T. Gervens and M. Schreiner (1998), *Constructive Approximation on the Sphere*, Clarendon Press, Oxford.

W. Freeden, M. Schreiner and R. Franke (1997), 'A survey on spherical spline approximation', *Surv. Math. Ind.* **7**, 29–85.

T.-P. Fries and H.-G. Matthies (2004), Classification and overview of meshfree methods. Informatikbericht Nr. 2003-3, Scientific Computing, Universität Braunschweig: `http://opus.tu-bs.de/opus/volltexte/2003/418/`.

P. Giesl (2005), Construction of global Lyapunov functions using radial basis functions, Habilitationsschrift, Technische Universität München.

J. Glaunés, M. Vaillant and M. I. Miller (2004), 'Landmark matching via large deformation diffeomorphisms on the sphere', *J. Math. Imaging Vis.* **20**, 179–200.

M. A. Golberg and C. S. Chen (1999), The method of fundamental solutions for potential, Helmholtz and diffusion problems, in *Boundary Integral Methods: Numerical and Mathematical Aspects*, Vol. 1 of *Comput. Eng.*, WIT Press/Comput. Mech. Publ., Boston, MA, pp. 103–176.

M. A. Golberg, A. S. Muleshkov, C. S. Chen and A. H.-D. Cheng (2003), 'Polynomial particular solutions for certain kinds of partial differential operators', *Numer. Methods Partial Differential Equations* **19**, 112–133.

M. v. Golitschek and W. A. Light (2001), 'Interpolation by polynomials and radial basis functions on spheres', *Constr. Approx.* **17**, 1–18.

J. Gomes, L. Darsa, B. Costa and L. Velho (1998), *Warping and Morphing of Graphical Objects*, Morgan Kaufmann, San Francisco, CA, USA.

J. J. Green (2002), Approximation with the radial basis functions of Lewitt, in *Algorithms for Approximation IV* (J. Levesley, I. Anderson and J. C. Mason, eds), University of Huddersfield, pp. 212–219.

L. Greengard and V. Rokhlin (1987), 'A fast algorithm for particle simulations', *J. Comput. Phys.* **73**, 325–348.

L. Greengard and J. Strain (1991), 'The fast Gauss transform', *SIAM J. Sci. Statist. Comput.* **12**, 79–94.

T. Gutzmer (1996), 'Interpolation by positive definite functions on locally compact groups with application to SO(3)', *Result. Math.* **29**, 69–77.

S. J. Hales and J. Levesley (2002), 'Error estimates for multilevel approximation using polyharmonic splines', *Numer. Algorithms* **30**, 1–10.

R. L. Harder and R. N. Desmarais (1972), 'Interpolation using surface splines', *J. Aircraft* **9**, 189–197.

T. Hastie, R. Tibshirani and J. Friedman (2001), *The Elements of Statistical Learning*, Springer Series in Statistics, Springer, New York.

Y. C. Hon and R. Schaback (2001), 'On unsymmetric collocation by radial basis functions', *Appl. Math. Comput.* **119**, 177–186.

Y. C. Hon and T. Wei (2002), A meshless computational method for solving inverse heat conduction problem, in *Boundary Elements XXIV* (C. Brebbia, ed.), WIT Press, pp. 135–144.

Y. C. Hon and T. Wei (2003), A meshless scheme for solving inverse problems of Laplace equation, in *Recent Development in Theories and Numerics*, World Scientific, River Edge, NJ, pp. 291–300.

Y. C. Hon and T. Wei (2004), 'A fundamental solution method for inverse heat conduction problem', *Engrg. Anal. Boundary Elements* **28**, 489–495.

Y. C. Hon and T. Wei (2005), 'The method of fundamental solution for solving multidimensional inverse heat conduction problems', *CMES Comput. Model. Eng. Sci.* **7**(2), 119–132.

Y. C. Hon and Z. Wu (2000*a*), 'Additive Schwarz domain decomposition with a radial basis approximation', *Internat. J. Appl. Math.* **4**, 81–98.

Y. C. Hon and Z. Wu (2000*b*), 'A numerical computation for inverse boundary determination problems', *Engrg. Anal. Boundary Elements* **24**, 599–606.

Y. C. Hon, R. Schaback and X. Zhou (2003), 'An adaptive greedy algorithm for solving large RBF collocation problems', *Numer. Algorithms* **32**, 13–25.

H. Hoppe (1994), Surface reconstruction from unorganized points. PhD thesis, University of Washington.

H. Hoppe, T. DeRose, T. Duchamp, J. McDonald and W. Stuetzle (1992), 'Surface reconstruction from unorganized points', *Computer Graphics* (*Proc. SIGGRAPH'92*) **26**, 71–78.

L. Hörmander (2003), *The Analysis of Linear Partial Differential Operators I: Distribution Theory and Fourier Analysis*, Classics in Mathematics, Springer.

S. Hubbert and T. M. Morton (2004), 'L_p-error estimates for radial basis function interpolation on the sphere', *J. Approx. Theory* **129**, 58–77.

M. S. Ingber, C. S. Chen and J. A. Tanski (2004), 'A mesh free approach using radial basis functions and parallel domain decomposition for solving three-dimensional diffusion equations', *Internat. J. Numer. Methods Engrg.* **60**, 2183–2201.

A. Iske (1995), Reconstruction of functions from generalized Hermite–Birkhoff data, in *Approximation Theory VIII*, Vol. 1 (C. Chui and L. Schumaker, eds), World Scientific, Singapore, pp. 257–264.

A. Iske (2003), 'Radial basis functions: basics, advanced topics and meshfree methods for transport problems', *Rend. Sem. Mat. Univ. Pol. Torino* **61**, 247–285.

A. Iske (2004), *Multiresolution Methods in Scattered Data Modelling*, Vol. 37 of *Lecture Notes in Computational Science and Engineering*, Springer, Berlin.

A. Iske and T. Sonar (1996), 'On the structure of function spaces in optimal recovery of point functionals for ENO-schemes by radial basis functions', *Numer. Math.* **74**, 177–201.

K. Jetter, J. Stöckler and J. Ward (1999), 'Error estimates for scattered data interpolation on spheres', *Math. Comput.* **68**, 733–747.

E. J. Kansa (1986), Application of Hardy's multiquadric interpolation to hydrodynamics, in *Proc. 1986 Simul. Conf.*, pp. 111–117.

I. Kovačevič, A. Poredoš and B. Šarler (2003), 'Solving the Stefan problem with the radial basis function collocation method', *Numerical Heat Transfer, Part B: Fundamentals* **44**, 575–599.

P. K. Kythe (1996), *Fundamental Solutions for Differential Operators and Applications*, Birkhäuser, Boston, MA.

A. La Rocca, A. Hernandez Rosales and H. Power (2005), ‘Radial basis function Hermite collocation approach for the solution of time dependent convection-diffusion problems’, *Engrg. Anal. Boundary Elements* **29**, 359–370.

P. Lancaster and K. Salkauskas (1981), ‘Surfaces generated by moving least squares methods’, *Math. Comput.* **37**, 141–158.

F. Lanzara, V. Maz’ya and G. Schmidt (2005), Approximate approximations from scattered data. WIAS Preprint No. 1058.

E. Larsson and B. Fornberg (2003), ‘A numerical study of some radial basis function based solution methods for elliptic PDEs’, *Comput. Math. Appl.* **46**, 891–902.

E. Larsson and B. Fornberg (2005), ‘Theoretical and computational aspects of multivariate interpolation with increasingly flat radial basis functions’, *Comput. Math. Appl.* **49**, 103–130.

S.-H. Lee and Y.-C. Yoon (2004), ‘Meshfree point collocation method for elasticity and crack problems’, *Internat. J. Numer. Methods Engrg.* **61**, 22–48.

J. Levesley and A. K. Kushpel (1999), ‘Generalised sk-spline interpolation on compact abelian groups’, *J. Approx. Theory* **97**, 311–333.

J. Levesley, Y. Xu, W. A. Light and W. E. Cheney (1996), ‘Convolution operators for radial basis approximation’, *SIAM J. Math. Anal.* **27**, 286–304.

D. Levin (1999), ‘Stable integration rules with scattered integration points’, *J. Comput. Appl. Math.* **112**, 181–187.

R. Lewitt, S. Matej and G. Herman (1997), Discretization and iterative solution of inverse problems in 3D computed tomography using bell-shaped radial basis functions having compact support. Technical report, Medical Image Processing Group, Dept. of Radiology, University of Pennsylvania, Philadelphia.

J. Li and C. S. Chen (2003), ‘Some observations on unsymmetric radial basis function collocation methods for convection-diffusion problems’, *Internat. J. Numer. Methods Engrg.* **57**, 1085–1094.

J. Li and Y. C. Hon (2004), ‘Domain decomposition for radial basis meshless methods’, *Numer. Methods Partial Differential Equations* **20**, 450–462.

S. Li and W. K. Liu (2002), ‘Meshfree and particle methods and applications’, *Appl. Mech. Rev.* **55**, 1–34.

S. Li and W. K. Liu (2004), *Meshfree Particle Methods*, Springer, Berlin.

X. Li (2005), ‘On convergence of the method of fundamental solutions for solving the Dirichlet problem of Poisson’s equation’, *Adv. Comput. Math.* **23**, 265–277.

W. A. Light and H. Wayne (1998), ‘On power functions and error estimates for radial basis function interpolation’, *J. Approx. Theory* **92**, 245–266.

L. Ling (2005), ‘Multidimensional quasi-interpolation formula with dimension-splitting multiquadric basis’, *Appl. Math. Comput.* **161**, 195–209.

L. Ling and E. J. Kansa (2004), ‘Preconditioning for radial basis functions with domain decomposition methods’, *Math. Comput. Modelling* **40**, 1413–1427.

L. Ling and E. J. Kansa (2005), ‘A least-squares preconditioner for radial basis functions collocation methods’, *Adv. Comput. Math.* **23**, 31–54.

L. Ling and R. Schaback (2004), On adaptive unsymmetric meshless collocation, in *Proc. 2004 International Conference on Computational and Experimental*

Engineering and Sciences (S. N. Atluri and A. J. B. Tadeu, eds). Vol. CD-ROM, *Advances in Computational and Experimental Engineering and Sciences*, Tech Science Press, Forsyth, USA, paper # 270.

L. Ling and M. R. Trummer (2004), 'Multiquadric collocation method with integral formulation for boundary layer problems', *Comput. Math. Appl.* **48**, 927–941.

Y. Liu, K. M. Liew, Y. Hon and X. Zhang (2005), 'Numerical simulation and analysis of an electroactuated beam using a radial basis function', *Smart Materials and Structures* **14**, 1163–1171.

R. Lorentz, F. J. Narcowich and J. D. Ward (2003), 'Collocation discretization of the transport equation with radial basis functions', *Appl. Math. Comput.* **145**, 97–116.

S. Lowitzsch (2005), 'Matrix-valued radial basis functions: Stability estimates and applications.', *Adv. Comput. Math.* **23**, 299–315.

Z. Luo and J. Levesley (1998), 'Error estimates and convergence rates for variational Hermite interpolation', *J. Approx. Theory* **95**, 264–279.

D. H. McLain (1974), 'Drawing contours from arbitrary data points', *Comput. J.* **17**, 318–324.

D. H. McLain (1976), 'Two dimensional interpolation from random data', *Comput. J.* **19**, 178–181.

W. R. Madych and S. A. Nelson (1988), 'Multivariate interpolation and conditionally positive definite functions', *Approx. Theory Appl.* **4**, 77–89.

W. R. Madych and S. A. Nelson (1990), 'Multivariate interpolation and conditionally positive definite functions II', *Math. Comput.* **54**, 211–230.

W. R. Madych and S. A. Nelson (1992), 'Bounds on multivariate polynomials and exponential error estimates for multiquadric interpolation', *J. Approx. Theory* **70**, 94–114.

N. Mai-Duy and T. Tran-Cong (2001), 'Numerical solution of Navier–Stokes equations using radial basis function networks', *Internat. J. Numer. Meth. Fluids* **37**, 65–86.

S. de Marchi, R. Schaback and H. Wendland (2005), 'Near-optimal data-independent point locations for radial basis function interpolation', *Adv. Comput. Math.* **23**, 317–330.

R. Mathon and R. L. Johnston (1977), 'The approximate solution of elliptic boundary-value problems by fundamental solutions', *SIAM J. Numer. Anal.* **14**, 638–650.

V. Maz'ya (1994), Approximate approximations, in *The Mathematics of Finite Elements and Applications*, Wiley, Chichester, pp. 77–104.

V. Maz'ya and G. Schmidt (2001), 'On quasi-interpolation with non-uniformly distributed centers on domains and manifolds', *J. Approx. Theory* **110**, 125–145.

B. Mederos, L. Velho and L. H. de Figueiredo (2003), Moving least squares multiresolution surface approximation, in *XVI Brazilian Symposium on Computer Graphics and Image Processing* (*SIBGRAPI'03*), p. 19.

J. Melenk and I. Babuška (1996), 'The partition of unity finite element method: Basic theory and applications', *Comput. Methods Appl. Mech. Engrg.* **139**, 289–314.

H. Meschkowski (1962), *Hilbertsche Räume mit Kernfunktion*, Springer, Berlin.

C. A. Micchelli (1986), 'Interpolation of scattered data: Distance matrices and conditionally positive definite functions', *Constr. Approx.* **2**, 11–22.

C. A. Micchelli and T. J. Rivlin (1977), A survey of optimal recovery, in *Optimal Estimation in Approximation Theory* (C. A. Micchelli and T. J. Rivlin, eds), Plenum Press, pp. 1–54.

C. A. Micchelli, T. J. Rivlin and S. Winograd (1976), 'Optimal recovery of smooth functions', *Numer. Math.* **26**, 191–200.

J. J. Monaghan (2005), 'Smoothed particle hydrodynamics', *Reports on Progress in Physics* **68**, 1703–1759.

T. M. Morton and M. Neamtu (2002), 'Error bounds for solving pseudodifferential equations on spheres by collocation with zonal kernels', *J. Approx. Theory* **114**, 242–268.

C. T. Mouat (2001), Fast algorithms and preconditioning techniques for fitting radial basis functions. PhD thesis, Mathematics Department, University of Canterbury, Christchurch, New Zealand.

C. Müller (1966), *Spherical Harmonics*, Springer, Berlin.

F. J. Narcowich (1995), 'Generalized Hermite interpolation and positive definite kernels on a Riemannian manifold', *J. Math. Anal. Appl.* **190**, 165–193.

F. J. Narcowich and J. D. Ward (1991), 'Norms of inverses and condition numbers for matrices associated with scattered data', *J. Approx. Theory* **64**, 69–94.

F. J. Narcowich and J. D. Ward (1992), 'Norm estimates for the inverse of a general class of scattered-data radial-function interpolation matrices', *J. Approx. Theory* **69**, 84–109.

F. J. Narcowich and J. D. Ward (1994*a*), 'Generalized Hermite interpolation via matrix-valued conditionally positive definite functions', *Math. Comput.* **63**, 661–687.

F. J. Narcowich and J. D. Ward (1994*b*), 'On condition numbers associated with radial-function interpolation', *J. Math. Anal. Appl.* **186**, 457–485.

F. J. Narcowich and J. D. Ward (2004), 'Scattered data interpolation on spheres: Error estimates and locally supported basis functions', *SIAM J. Math. Anal.* **33**, 1393–1410.

F. J. Narcowich, R. Schaback and J. D. Ward (1999), 'Multilevel interpolation and approximation', *Appl. Comput. Harmon. Anal.* **7**, 243–261.

F. J. Narcowich, X. Sun and J. D. Ward (2006), 'Approximation power of RBFs and their associated SBFs: A connection', to appear in *Adv. Comput. Math.*

F. J. Narcowich, X. Sun, J. D. Ward and H. Wendland (2005*a*), Direct and inverse Sobolev error estimates for scattered data interpolation via spherical basis functions. Preprint, College Station, TX.

F. J. Narcowich, J. D. Ward and H. Wendland (2004), Sobolev error estimates and a Bernstein inequality for scattered data interpolation via radial basis functions. Preprint, College Station, TX. To appear in *Constr. Approx.*

F. J. Narcowich, J. D. Ward and H. Wendland (2005*b*), 'Sobolev bounds on functions with scattered zeros, with applications to radial basis function surface fitting', *Math. Comput.* **74**, 643–763.

D. Nardini and C. A. Brebbia (1982), A new approach to free vibration analysis using boundary elements, in *Boundary Element Methods in Engineering* (C. A. Brebbia, ed.), Springer, New York, pp. 312–326.

J. Y. Noh, D. Fidaleo and U. Neumann (2000), Animated deformations with radial basis functions, in *VRST '00: Proc. ACM Symposium on Virtual Reality Software and Technology*, ACM Press, New York, USA, pp. 166–174.

Y. Ohtake, A. Belyaev, M. Alexa, G. Turk and H.-P. Seidel (2003*a*), 'Multi-level partition of unity implicits', *ACM Trans. Graphics* **22**, 463–470.

Y. Ohtake, A. Belyaev and H.-P. Seidel (2003*b*), A multi-scale approach to 3D scattered data interpolation with compactly supported basis functions, in *Shape Modeling International*, IEEE Computer Society Press, pp. 153–164.

R. Opfer (2006), 'Multiscale kernels', to appear in *Adv. Comput. Math.*

P. Partridge, C. Brebbia and L. Wrobel (1992), *The Dual Reciprocity Boundary Element Method*, CMP/Elsevier.

D. W. Pepper and B. Šarler (2005), 'Application of meshless methods in thermal analysis', *Mechanical Engineering J.* **51**, 476–483.

A. Pinkus (2004), 'Strictly Hermitian positive definite functions', *J. d'Analyse Math.* **94**, 293–318.

T. Poggio and S. Smale (2003), 'The mathematics of learning: dealing with data', *Notices Amer. Math. Soc.* **50**, 537–544.

A. Poullikkas, A. Karageorghis and G. Georgiou (2002), 'The method of fundamental solutions for three-dimensional elastostatics problems', *Comput. & Structures* **80**, 365–370.

H. Power and V. Barraco (2002), 'A comparison analysis between unsymmetric and symmetric radial basis function collocation methods for the numerical solution of partial differential equations', *Comput. Math. Appl.* **43**, 551–583.

C. Rabut (1989), Fast quasi-interpolation of surfaces with generalized B-splines on regular nets, in *Mathematics of Surfaces III* (D. C. Handscomb, ed.), Clarendon Press, Oxford, pp. 429–449.

D. L. Ragozin (1983), 'Error bounds for derivative estimates based on spline smoothing of exact or noisy data', *J. Approx. Theory* **37**, 335–355.

P. Ramachandran and K. Balakrishnan (2000), 'Radial basis functions as approximate particular solutions: Review of recent progress', *Engrg. Anal. Boundary Elements* **24**, 575–582.

C. H. Reinsch (1967), 'Smoothing by spline functions', *Numer. Math.* **10**, 177–183.

C. H. Reinsch (1971), 'Smoothing by spline functions II', *Numer. Math.* **16**, 451–454.

G. Roussos (1999), Computation with radial basis functions, PhD thesis, Imperial College of Science Technology and Medicine, University of London.

B. Šarler (2005), 'A radial basis function collocation approach in computational fluid dynamics', *Comput. Methods Engineering Sci.* **7**, 185–193.

B. Šarler, J. Perko and C. S. Chen (2004), 'Radial basis function collocation method solution of natural convection in porous media', *Internat. J. Numer. Methods Heat Fluid Flow* **14**, 187–212.

R. Schaback (1995*a*), Creating surfaces from scattered data using radial basis functions, in *Mathematical Methods for Curves and Surfaces* (T. L. M. Daehlen and L. Schumaker, eds), Vanderbilt University Press, Nashville, TN, pp. 477–496.

R. Schaback (1995*b*), 'Error estimates and condition number for radial basis function interpolation', *Adv. Comput. Math.* **3**, 251–264.

R. Schaback (1997), On the efficiency of interpolation by radial basis functions, in *Surface Fitting and Multiresolution Methods* (A. LeMéhauté, C. Rabut and L. Schumaker, eds), Vanderbilt University Press, Nashville, TN, pp. 309–318.

R. Schaback (1999), Native Hilbert spaces for radial basis functions I, in *New Developments in Approximation Theory* (M. D. Buhmann, D. H. Mache, M. Felten and M. W. Müller, eds), Vol. 132 of *International Series of Numerical Mathematics*, Birkhäuser, pp. 255–282.

R. Schaback (2005*a*), Convergence analysis of methods for solving general equations, in *Boundary Elements XXVII* (A. Kassab, C. Brebbia, E. Divo and D. Poljak, eds), WIT Press, Southampton, pp. 17–24.

R. Schaback (2005*b*), Convergence of unsymmetric kernel-based meshless collocation methods. Preprint, Universität Göttingen.

R. Schaback and H. Wendland (2000*a*), 'Adaptive greedy techniques for approximate solution of large RBF systems', *Numer. Algorithms* **24**(3), 239–254.

R. Schaback and H. Wendland (2000*b*), Numerical techniques based on radial basis functions, in *Curve and Surface Fitting: Saint-Malo 1999* (A. Cohen, C. Rabut and L. L. Schumaker, eds), Vanderbilt University Press, Nashville, TN, pp. 359–374.

R. Schaback and J. Werner (2006), 'Linearly constrained reconstruction of functions by kernels with applications to machine learning', to appear in *Adv. Comput. Math.*

I. J. Schoenberg (1937), 'On certain metric spaces arising from Euclidean spaces by a change of metric and their imbedding in Hilbert space', *Ann. of Math.* **38**, 787–793.

I. J. Schoenberg (1942), 'Positive definite functions on spheres', *Duke Math. J.* **9**, 96–108.

B. Schölkopf and A. J. Smola (2002), *Learning with Kernels*, MIT Press, Cambridge.

J. Shawe-Taylor and N. Cristianini (2004), *Kernel Methods for Pattern Analysis*, Cambridge University Press.

D. Shepard (1968), A two dimensional interpolation function for irregularly spaced data, in *Proc. ACM National Conference*, pp. 517–524.

C. Shu, H. Ding and K. S. Yeo (2005), 'Computation of incompressible Navier–Stokes equations by local RBF-based differential quadrature method', *Comput. Methods Engrg. Sci.* **7**, 195–206.

S. Smale and D.-X. Zhou (2003), 'Estimating the approximation error in learning theory', *Anal. Appl.* (*Singap.*) **1**, 17–41.

A. J. Smola and B. Schölkopf (1998), 'On a kernel-based method for pattern recognition, regression, approximation and operator inversion', *Algorithmica* **22**, 211–231.

T. Sonar (1996), 'Optimal recovery using thin plate splines in finite volume methods for the numerical solution of hyperbolic conservation laws', *IMA J. Numer. Anal.* **16**, 549–581.

J. Stewart (1976), 'Positive definite functions and generalizations: An historical survey', *Rocky Mountain J. Math.* **6**, 409–434.

J. F. Traub and A. G. Werschulz (1998), *Complexity and Information*, Oxford University Press, Oxford, UK.

G. Turk and J. F. O'Brien (2002), 'Modelling with implicit surfaces that interpolate', *ACM Trans. Graphics* **21**, 855 – 873.

G. Wahba (1975), 'Smoothing noisy data by spline functions', *Numer. Math.* **24**, 383–393.

G. Wahba (1990), *Spline Models for Observational Data*, CBMS-NSF, Regional Conference Series in Applied Mathematics, SIAM, Philadelphia.

T. Wei, Y. Hon and Y. B. Wang (2005), 'Reconstruction of numerical derivatives from scattered noisy data', *Inverse Problems* **21**, 657 – 672.

H. Wendland (1995), 'Piecewise polynomial, positive definite and compactly supported radial functions of minimal degree', *Adv. Comput. Math.* **4**, 389–396.

H. Wendland (2001), 'Local polynomial reproduction and moving least squares approximation', *IMA J. Numer. Anal.* **21**, 285–300.

H. Wendland (2004), Solving large generalized interpolation problems efficiently, in *Advances in Constructive Approximation* (M. Neamtu and E. B. Saff, eds), Nashboro Press, Brentwood, TN, pp. 509–518.

H. Wendland (2005*a*), 'On the convergence of a general class of finite volume methods', *SIAM J. Numer. Anal.* **43**, 987–1002.

H. Wendland (2005*b*), *Scattered Data Approximation*, Cambridge Monographs on Applied and Computational Mathematics, Cambridge University Press, Cambridge, UK.

H. Wendland and C. Rieger (2005), 'Approximate interpolation with applications to selecting smoothing parameters', *Numer. Math.* **101**, 643–662.

Z. Wu (1992), 'Hermite–Birkhoff interpolation of scattered data by radial basis functions', *Approx. Theory Appl.* **8**, 1–10.

Z. Wu (1995), 'Multivariate compactly supported positive definite radial functions', *Adv. Comput. Math.* **4**, 283–292.

Z. Wu and R. Schaback (1993), 'Local error estimates for radial basis function interpolation of scattered data', *IMA J. Numer. Anal.* **13**, 13–27.

D.-X. Zhou (2002), 'The covering number in learning theory', *J. Complexity* **18**, 739–767.

X. Zhou, Y. C. Hon and J. Li (2003), 'Overlapping domain decomposition method by radial basis functions', *Appl. Numer. Math.* **44**, 241–255.