Approximation of scattered data using smooth grid functions

Erlend Arge*, Morten Dæhlen, Aslak Tveito

SINTEF-SI, P.O. Box 124, Blindern, 0314 Oslo 3, Norway

Received 19 July 1993; revised 14 January 1994

Abstract

We present a three-stage scheme for constructing smooth grid functions approximating data defined over scattered point sets in $\mathbb{R}^s$. The scheme is useful for approximating large scattered data sets and is particularly successful when the data points are unevenly distributed. The paper includes several examples of grid surface construction over the plane.

Keywords: Scattered data approximation; Regular grids; Smoothing techniques

1. Introduction

Grid functions, i.e. functions defined on the nodes of regular grids, are common means for representing surfaces in many branches of the technical sciences. Examples are terrain modeling in the mapping industries and the modeling of geological surfaces in oil reservoir engineering.

Despite their limitations and simplistic nature, engineers often prefer to use grid functions because they are simple to understand, represent, manipulate and visualize on a digital computer. In addition, it is easy to extend grid functions to continuous functions defined on $\mathbb{R}^s$, for which there exist numerous methods, e.g. spline interpolation, cubic Hermite interpolation, bilinear interpolation and box spline interpolation [3].

Given a regular grid, a central problem is to construct a grid function that approximates an underlying set of scattered measurement data. The process of generating grid functions approximating scattered data is often referred to as gridding. Many methods for gridding proceed in a local fashion, in the sense that the value assigned to a grid node is computed from a limited number of neighboring scattered data points, see for example [5]. The local approximation schemes used are often standard methods such as Shepard's method [8], radial basis functions [7], least-squares methods or triangulations. We refer to [9], for which the total volume is devoted to gridding.

* This research was partially supported by the The Royal Norwegian Council for Scientific and Industrial Research.

* Corresponding author.

0377-0427/95/$09.50 © 1995 Elsevier Science B.V. All rights reserved

SSDI 0377-0427(94)00033-W
It is important to notice that scattered measurement data often are unevenly distributed in the argument domain, as is for example the case with contour data and seismic track data. Such features of the data will in most cases be problematic for methods based only on local approximation schemes. The reason is that values of grid nodes in regions without data will be determined from distant data values. This can be unstable and lead to unpredictable results and unwanted geometric properties.

In the present paper we propose a method for gridding based on local as well as global approximation schemes. The key idea is to use a local approach to assign values to grid nodes in areas where the data density is high, and then extend these values to the entire grid via a global extrapolation technique.

The scheme can be divided into three main steps:

1. Regularization. From the given scattered data set determine a subset $D$ of grid nodes lying in regions where the data density is high.
2. Approximation. Determine values at the nodes in $D$ using an approximant to some nearby data values.
3. Extrapolation. Extend the values defined on $D$ to the entire grid.

The data set constructed by the first two steps of the above procedure will be said to be regularly scattered, as it consists of scattered data lying on a subset of a regular grid. In [2] it was shown how one can construct box spline interpolants to regularly scattered data sets in $\mathbb{R}^s$ by the use of a certain global smoothing operator. This gives well-behaved approximations in regions with few data points, which is problematic when using only local methods.

The extrapolation technique in Step 3 of the present method follows the same line of thought. However, in addition to requiring the surface to be well behaved in regions with little data, we show how one can construct smoothing operators to impose shape constraints on the surface, and thereby act as an active tool in the process of designing the approximation.

The use of global extrapolation techniques leads to the problem of solving potentially large linear systems of equations for the unknown values at the grid nodes. Due to the regularity of the grid and the realization of the smoothing operators as grid point operators, these linear systems are well suited for being solved with iterative methods. Thus, questions regarding convergence, nonsingularity, symmetry and positive definiteness becomes important. The answers to these questions depend on the choice of grid point operators as well as configuration of the unknown grid nodes. We will, however, not go into detail on these issues in the present paper. Rather, the objective here is to demonstrate some possibilities with this approach to gridding, and to point out some problem areas which should to be addressed in order to improve the method.

In the examples we will focus on gridding in two variables. However, it should be noted that the method is equally easy to implement in a general number of variables, which could make the method attractive as a general tool. We refer to Alfeld [1] for a survey of some methods for scattered data approximation in three or more variables.

We start in Section 2 by discussing the regularization and the approximation process. The extrapolation process is discussed in Section 3, and in Section 4 the method is illustrated with several examples of grid surface construction over the plane. We close the paper with some concluding remarks.
2. Regularization and approximation

In this section we will present the basic principles of the regularization and the approximation phase. More details and specific choices are presented in the examples of Section 3.

Given a set of scattered data $S \times Y = (s_k, y_k)_{k \in K} \subset \mathbb{R}^s \times \mathbb{R}$ for $k$ in some finite index set $K$, we consider the problem of transforming this data set into a regularly scattered data set $(i, f_i)$, $i \in D$, where $D$ is a subset of a regular grid. The basic idea is to estimate a value at a grid node only if the density of scattered points surrounding the node is sufficiently high. The values at the remaining nodes are estimated by the extrapolation technique presented in Section 3.

Let $\Omega \subset \mathbb{R}^s$ be a bounded domain containing $S$. Let $R = \Omega \cap h\mathbb{Z}^s$, i.e. $R$ is the set of nodes in $\Omega$ of a uniform grid in $\mathbb{R}^s$ with grid spacing $h$. Assume further that a threshold value $\delta \geq 0$ and a density function $\phi(S|\cdot): \Omega \rightarrow [0, \infty)$, depending on $S$, is given. We then define

$$D = \{i \in R: \phi(S|i) > \delta\},$$

i.e. $D$ is the subset of $R$ for which the density of scattered points exceeds the threshold value. The nodes in $R$ not in $D$ is denoted by $E$. We note that we use the convention of writing $i \in R$, even if $i$ is of the form $i = hk$ for some $k \in \mathbb{Z}^s$.

We have experimented with two different types of density functions, defined by

$$h_{\phi_2}(S|v) = \frac{h}{2 \min_{s \in S} ||s - v||_2}$$

and

$$h_{\phi_\infty}(S|v) = \frac{h}{2 \min_{s \in S} ||s - v||_\infty}.$$ 

Here, $||\cdot||_2$ and $||\cdot||_\infty$ denote the usual $l_2$ and $l_\infty$ norms, respectively. The density functions are continuous on $\Omega \setminus S$ and take on the value $\infty$ on $S$.

In Fig. 1 we have illustrated the set $D$ defined on the basis of $\phi_\infty$ and threshold value $\delta = 1$. The reason for using the particular size of $\delta$ is that this is the smallest possible value making sure that every point in $S$ has a neighboring grid point in $D$. Decreasing $\delta$ will add more points to $D$, and since the value of the grid function at the nodes of $D$ is to be determined through local approximation, smaller size of $\delta$ put more emphasis on the local than the global approximation scheme.

There are two main issues connected to determining the values on $D$. These are the choice of local approximation scheme and the selection of point sets determining the local approximation. To be more precise, for each $i \in D$ we must choose a local approximation operator $L_i: \mathbb{R}^s \rightarrow \mathbb{R}$ defined from a chosen subset $S_i \times Y_i \subset S \times Y$ of the scattered data, and sample $L_i$ at $i$ for the value.

Examples of $L_i$'s are approximation to $S_i \times Y_i$ through Shepard's method, the radial basis function method or polynomial least-squares methods, which are all easy to construct for a general number of variables. In some situations one might also use different methods for different $i$'s, for example could one use least squares for some nodes and Shepard's method for others.
A successful construction of the local approximation operator $L_i$ will often involve requirements connected to the position and number of points in $S_i$. For example, using least squares with quadratic polynomials in two space dimensions, the number of points in $S_i$ must be at least 7 and they must not lie on a conic section. In addition, depending on the type of data one might impose additional constraints on the distribution of points in $S_i$, mainly in order to capture local trends in the data. We will come back to such considerations in the examples.

3. Extrapolation

For the purpose of defining global extrapolating schemes, it is convenient to introduce the notion of grid functions and grid point operators.

Thus, we interpret a vector $x = (x_j)_{j \in \mathbb{Z}^s}$ as a grid function $x : \mathbb{Z}^s \to \mathbb{R}$. For any subset $J \subset \mathbb{Z}^s$, we let $G(J)$ be the space of grid functions which are supported in $J$, i.e.

$$x \in G(J) \iff x_j = 0 \quad \forall j \in J.$$

A grid point operator $Q : G(\mathbb{Z}^s) \to G(\mathbb{Z}^s)$ is often identified with a grid function, or a mask, of finite support, i.e. $Q = (Q_i)_{i \in \mathbb{Z}^s}$, and operates on an element $x \in G(\mathbb{Z}^s)$ as a discrete convolution

$$(Qx)_j = \sum_{i \in \mathbb{Z}^s} Q_i x_{j-i}.$$

It will, however, for our purpose be convenient to allow the mask to vary over the domain of interest. Thus, we will consider grid point operators defined through a rule of the form

$$(Qx)_j = \sum_{i \in \mathbb{Z}^s} Q_i(j) x_{j-i}.$$
The mask connected to a given \( j \in \mathbb{Z}^s \) will be denoted \( Q(j) \). This makes it possible to construct operators that take into account special features of a particular approximation problem.

The adjoint of a grid point operator \( Q \) with respect to the Euclidean inner product is the unique grid point operator \( Q^t \) satisfying

\[
(Qx,y) = (x,Q^t y)
\]

for all \( x,y \in G(\mathbb{Z}^s) \). Here

\[
(x,y) = \sum_{j \in \mathbb{Z}^s} x_j y_j.
\]

In order to compute the elements of the adjoint, let \( x,y \in G(\mathbb{Z}^s) \) and assume that \( x \) and \( y \) have finite support. Then

\[
(Qx,y) = \sum_{j \in \mathbb{Z}^s} \sum_{i \in \mathbb{Z}^s} Q_i(j)x_{j-i} y_j = \sum_{i \in \mathbb{Z}^s} \sum_{j \in \mathbb{Z}^s} Q_i(j+i)x_j y_{j+i} = \sum_{j \in \mathbb{Z}^s} \sum_{i \in \mathbb{Z}^s} Q_{-i}(j-i)x_j y_{j-i} = \sum_{j \in \mathbb{Z}^s} Q_{-i}(j-i)x_{j-i} y_j.
\]

This implies that the adjoint \( Q^t \) satisfies

\[
Q^t(j) = Q_{-i}(j - i).
\]

If \( Q_i(j) = Q_i(j) \) for all \( i \) and \( j \) then \( Q \) is self-adjoint with respect to the Euclidean inner product. In particular if \( Q \) does not vary with \( j \) then \( Q \) is self-adjoint if and only if \( Q_i = Q_{-i} \) for all \( i \) or equivalently if the mask is symmetric around the origin. If \( Q(j) \) vary over the domain, then \( Q \) may still be self-adjoint. However, the mask \( Q(j) \) may fail to be symmetric around the origin.

Given a subspace \( G(J) \subset G(\mathbb{Z}^s) \), then \( Q \) is positive definite on \( G(J) \) if for every nonzero \( x \in G(J) \) we have \( (Qx,x) > 0 \).

We are now in position to address the extrapolation problem. We start by partitioning \( R = \Omega \cap \mathbb{Z}^s \) into two subsets \( D \) and \( E \), where \( D \), as constructed in Section 2, is the subset of nodes where there are given function values, and \( E \) is the subset where no values are given, cf. Fig. 1. As before, we assume that \( \Omega \) is bounded so that \( D \) and \( E \) are finite sets. The extrapolation problem can then be rephrased by saying that we are given a grid function \( \tilde{x} \in G(D) \) and we want to construct a grid function \( x \in G(E) \), which extrapolates \( \tilde{x} \) "naturally" to \( E \).

Let \( Q \) be a grid point operator defined on \( G(E) \), and let \( z \in G(E) \) be given. We then define the extrapolation \( x \) of \( \tilde{x} \in G(D) \) to \( E \) as the solution to a linear system of equations of the following form:

\[
(Qx)_j = z_j \quad \forall j \in E.
\]

Evidently, the quality of the extrapolation depends on the choice of the grid point operator \( Q \) and on the right-hand side \( z \). We will come back to specific choices of these in the examples below.
We will always use masks $Q(j)$ that have small supports, thus the coefficient matrix associated with (5) will be sparse. Since in addition the system efficiently can be expressed in terms of grid point operators, it is well suited for being solved with iterative solvers.

In the examples below the operator $Q$ is chosen in order to smooth the data, while $z$ depends on the given data values defined on $D$.

4. Examples

In this section we will present some examples of the use of the three-step gridding method described above. In the first two examples all steps of the method are illustrated, and the results are compared to a method based only on local approximations. In the rest of the examples the focus is on the extrapolation phase of the gridding, and we illustrate how the extrapolation operator can be chosen in order to handle certain shape constraints. In all examples the operators in question are self-adjoint and positive definite, and the corresponding systems are therefore solved with the conjugate gradient method which is a suitable solver under these circumstances (cf. [6]).

4.1. Dense and sparse data

We will illustrate here the three-step gridding with two examples in $\mathbb{R}^2$, one with a dense and one with a sparse data set. For both examples the three steps are constructed as described below.

Regularization: Using notation as in Section 2, the regularization process constructs the subset $D$ of the regular grid $R$ using (1) with $\delta = 1$ and the density function $\Phi_\infty$ defined in (3). We also experimented with the density function $\Phi_2$ defined in (2), however the results produced by $\Phi_\infty$ turned out to be better.

Approximation: In order to construct the subset $S_i$ of the scattered data set $S$ associated with a node $i \in D$, we use the basic approach to collect points from each of the four quadrants of the plane originated at $i$. The reason to do this is firstly to be able to capture the trend in the data. Secondly, we want to use a local approximation scheme with linear precision, and this requires that $S_i$ contains at least three points not on a line.

The precise construction is the following. Let

$$F_j = \{u \in \mathbb{R}^2 : \|u - i\|_\infty < jh\},$$

let $K^1, \ldots, K^4$ be the four closed quadrants originated at $i$ and defined $K^l_j = F_j \cap K^l$. Then $K^l_j$ is a square with side $jh$ extending into the quadrant $K^l$ with one corner at $i$. Define

$$j_i = \begin{cases} 0 & \text{if } K^l_i \cap S = \emptyset; \\ \min j : K^l_j \cap S \neq \emptyset & \text{otherwise.} \end{cases}$$

We can now define

$$S_i = (K^1_{j_i} \cup K^2_{j_i} \cup K^3_{j_i} \cup K^4_{j_i}) \cap S.$$
In most cases this definition of $S_i$ will give a good distribution of points around $i$, in the sense that all $K_{j_i}$'s have interior points. However, in some cases, especially in regions close to the boundary of the data, some of the sets $K_{j_i}$ can be empty. For this reason we have used two types of local approximants:

- linear precision Shepard's method if $S_i$ contains at least three points not on a line;
- ordinary Shepard's method otherwise.

**Extrapolation:** For a given spacing $h$, let $d_1 = he^1$ and $d_2 = he^2$, where $e^1$ and $e^2$ are the unit coordinate vectors in $\mathbb{R}^2$. We will use an operator $Q^1$ which in its basic form is defined for $j \in E$ by

$$
(Q^1 x)_j = 12x_j
- 4[x_{j-d_1} + x_{j+d_1} + x_{j-d_2} + x_{j+d_2}]
+ [x_{j-2d_1} + x_{j+2d_1} + x_{j-2d_2} + x_{j+2d_2}].
$$

(6)

However, for $j$ near the boundary of $R$ the above definition of the mask will access undefined data, and hence the mask must be redefined in such situations. Thus, if $j + 2d_1 \notin R$ we redefine the mask by setting

$$
x_{j+2d_1} = 2x_{j+d_1} - x_j,
$$

and if also $j + d_1 \notin R$ we redefine the mask by setting

$$
x_{j+d_1} = 2x_j - x_{j-d_1}.
$$

These redefinitions of the mask correspond to setting the second-order derivatives in the $d_1$ direction at $j$ and $j + 1$ to zero in a discrete sense. Similar redefinitions can be made in the other directions if necessary.

Now, let $\tilde{x} \in G(D)$ be the result of the local approximation process as defined in Step 2. Define $z \in G(E)$ by

$$
z_j = -(Q^1 \tilde{x})_j \quad \forall j \in E.
$$

(7)

Using (7) we can solve (5) for the unknown grid function $x \in G(E)$.

Under suitable hypothesis on the distribution of nodes in $E$ it can be shown that $Q^1$ is a symmetric and positive-definite operator on $G(E)$. We will not go into detail on these issues here, however, for both examples below these hypothesis are satisfied, and we have therefore solved the corresponding linear systems using the conjugate gradient method.

One can also show that solving (5) is equivalent to minimizing the sum

$$
\sum_{i \in E} \{(x + \tilde{x})^i_1 \}^2 + \{(x + \tilde{x})^i_2 \}^2
$$

over all $x \in G(E)$. Here

$$
y^i_1 = y_{i-d_1} - 2y_i + y_{i+d_1},
$$

$$
y^i_2 = y_{i-d_2} - 2y_i + y_{i+d_2}.
$$
Hence, the selected extrapolation of $\bar{x}$ to $E$ is the smoothest possible in the sense of minimizing the given second-order differences.

Contour data: In our first experiment we construct a geological surface given by scattered data as contour lines. The argument values of the data are shown in Fig. 2. The data set consists of 43 contour curves with a total of 5656 points.

Using a grid with $3721 = 61 \times 61$ nodes, the regularization process determined the subset $D$ with 1867 nodes located in regions close to the contour lines. Hence, in this example over 50% of the nodes in $R$ was determined with the local approximation scheme, which reflects that the scattered data set is relatively dense in relation to the chosen grid spacing.

We note that due to the construction of the local approximations, at least two neighbouring contours are in most cases used to determine the value at a grid node in $D$. This is important in order to capture the slope of the terrain.

In the extrapolation process we terminated the conjugate gradient algorithm when there were no visible differences between the iterates. In this example the algorithm was terminated after 185 iterations. In Section 5 we will study the convergence more closely.

The result of the three-step gridding is shown at the top of Fig. 3. At the bottom of the figure we have shown the result of applying the local approximation scheme to every node in $R$, i.e. with $D = R$. The result is not as smooth as the result of the three-step gridding, but is nevertheless acceptable. The reason why the difference between the results is moderate is that the size of $D$ in the three-step gridding is relatively large, due to the density of data and resolution of the grid. With a finer grid the situation is different as is illustrated in the next example.

Fine grid: In this example we have used a fine grid compared to the number of scattered data points. The grid is still of size $3721 = 61 \times 61$, however, the number of scattered points is only 24.

The regularization process determined $D$ with 24 points, for which the values were computed via the local approximation process. The conjugate gradient algorithm in the extrapolation phase was stopped after 2122 iterations. At this point there were no visible difference between the iterates. The reason why we had to use more iterations than in the proceeding example is because there are more unknowns, and because there are large areas without data.
Fig. 3. The result of gridding the contour data. The result of three-step gridding (top) and the result based on local approximations (bottom).

Fig. 4. The result of gridding a sparse data set. The result of three-step gridding (top) and the result based on local approximations (bottom). The scattered data points are superimposed on the surfaces.
Fig. 5. The result of extrapolating the function $f(u,v) = 1 - |u - v|$ using the operator defined in (6).

The result is shown at the top of Fig. 4. At the bottom of the figure we have shown the result of using the local approximation scheme at every node of the grid. In this case the differences between the results are apparent. As can be seen, discontinuities appear in the lower figure at the border lines of different local approximants. This behavior is typical when using only local approximation, but is less profound when the data density is high.

4.2. Operator design

In Fig. 5 we have shown the result of using the operator $Q^1$ defined in (6) for extrapolating a function with a ridge. For a given grid spacing $h$, the grid $R$ consists of all grid nodes in the unit square, $E \subset R$ consists of all grid nodes in $[\frac{1}{4}, \frac{1}{2}] \times [\frac{1}{4}, \frac{1}{2}]$, and $D = R \setminus E$. We note that the extrapolation may not be satisfactory in this case since it does not preserve the ridge.

The data given as boxes in Fig. 5 shows the grid function $\tilde{G}(D)$ which is obtained by sampling the function $f(u,v) = 1 - |u - v|$ on $D$. The ridge in $f$ is in the direction $d^3 = d^1 + d^2$, where $d^1$ and $d^2$ are defined in the previous example.

With $Q^1$ as defined in (6) and $z$ as in (7) we then solved (5) for the extrapolation grid function $\tilde{x} \in G(E)$. The number of grid nodes is in this case $40 \times 40 = 1600$, and the number of nodes in $E$ is 400.

We can preserve the ridge by choosing another operator $Q^2$. To this end, let

$$ (Q^2 x)_i = x_{i-2d^3} - 4x_{i-d^3} + 6x_i - 4x_{i+d^3} + x_{i+2d^3}, $$

for $i \in \mathbb{Z}^2$. This operator has smoothing effect only in the direction $d^3$ and not in the other directions, hence we would expect it to preserve edges in the direction $d^3$.

Again $z \in G(E)$ is defined as in (7). The result solving (5) using a uniform grid with 400 unknown nodes is shown in Fig. 6. We remark that the extrapolation in fact reconstructs the ridge in $f$. 
We also note that solving (5) in this case is equivalent to minimizing the sum
\[ \sum_{i \in \mathcal{E}} [(x + \bar{x})^{\frac{3}{2}}]^2, \]
over all \( x \in \mathcal{G}(\mathcal{E}) \) where \( y_i^{\frac{3}{2}} = y_{i-d^3} - 2y_i + y_{i+d^3}. \)

### 4.3. Blended operators

In the examples of the previous subsection we have used different operators motivated from different properties of the approximation problems. Of course, the particular properties of both these examples can occur in a single approximation problem. Thus, let us now consider an example where this is the case, leading to the definition of a grid point operator composed of multiple masks.

We consider the domain \( \Omega = [0, 2] \times [0, 1] \) and a function \( f \) defined on \( \Omega \) by
\[ f(u, v) = 1 - |u - v| + (u - 1)^3 \sin(4\pi v)\sin(4\pi u), \]
where \( y^3 = \max(y, 0)^3. \)

Let \( \Omega_2 \subset \Omega \) be the unit square \( [0, 1] \times [0, 1] \) and \( \Omega_1 \) be the remaining part of \( \Omega \). Then the function \( f \) coincides with the ridge function in the previous subsection on \( \Omega_2 \) with a ridge along the line \( u = v \). On \( \Omega_1, f \) oscillates smoothly. It is easily verified that \( f \) is \( C^2 \) at every point in \( \Omega \) not lying at the ridge \( u = v. \)

For a given grid spacing \( h \) let \( R = h\mathbb{Z}^2 \cap \Omega \), let \( E = R \cap [0.3, 1.7] \times [0.3, 0.7] \) and let \( D = R \setminus E \). The grid function \( \bar{x} \in G(D) \) is obtained by sampling \( f \) at the nodes of \( D \).

In order to define the grid point operator for this problem, we note that we would like it to behave like \( Q^2 \) defined in (6) in \( \Omega_2 \) and like \( Q^1 \) defined in (8) in \( \Omega_1 \). This motivates the definition of a grid point operator
\[ P(j) = \alpha(j)Q^1 + (1 - \alpha(j))Q^2, \]
where
\[ \alpha : \mathbb{R}^2 \rightarrow [0, 1] \]
is a function designed to construct a blend of $Q^1$ and $Q^2$. In our example we have used a piecewise cubic blend function of the form

$$ a(u,v) = \begin{cases} 
0 & \text{if } u \leq 1 - \omega, \\
3 \left( \frac{u - (1 - \omega)}{2\omega} \right)^2 - 2 \left( \frac{u - (1 - \omega)}{2\omega} \right)^3 & \text{if } u \in (1 - \omega, 1 + \omega), \\
1 & \text{if } u \geq 1 + \omega,
\end{cases} $$

where $\omega \geq 0$ is a parameter determining a blend region $[1 - \omega, 1 + \omega] \times [0, 1] \subset \Omega$.

It can be shown that the operator $P$ is positive definite. However, it is easily verified that $P$ does not satisfy (4) which implies that $P$ is not self-adjoint. Eq. (4) fails to hold when $j$ lies in a neighborhood of the blend region. It does, however, hold for $j$ away from this region.

Since $P$ is not self-adjoint we cannot solve the problem

$$ (Px)_j = - (P\bar{x})_j, \quad j \in E $$

using the conjugate gradient method. Even if other solvers apply to this problem (cf. [4]), we will instead define a new self-adjoint and positive-definite operator related to $P$ which makes the conjugate gradient method applicable.

It is well known that if $P$ is self-adjoint and positive definite then the solution to (10) minimizes the quadratic form

$$ \psi(x) = (Px + P\bar{x}, x + \bar{x}) $$

over all $x \in G(E)$. Based on the assumption that the solution to (10) is close to the minimum of $\psi$ we will instead of solving (10) compute the minimum of $\psi$.

In order to minimize $\psi$ define

$$ Q^3 = \frac{1}{2}(P + P^t), $$

where $P^t$ is the adjoint of $P$ defined by $P^t(j) = P(j - i)$. It is easily seen that $Q^3$ is self-adjoint. Moreover, we also have $(Q^3y, y) = (Py, y)$ for all grid functions $y$, and therefore

$$ \psi(x) = (Px + P\bar{x}, x + \bar{x}) = (Q^3x + Q^3\bar{x}, x + \bar{x}). $$

This implies that minimizing $\psi$ over $G(E)$ is equivalent to solving the linear system

$$ (Q^3x)_j = z_j, \quad j \in E, $$

with $z_j = - (Q^3\bar{x})_j$. Since $Q^3$ is self-adjoint and positive definite we can solve (11) using the conjugate gradient method.

The result solving (11) on a uniform grid with 896 unknown nodes is shown in Fig. 7. The solution varies smoothly over the blend region, it preserves the ridge and is smooth in the domain $\Omega_2$.

We note that good choices of the blend region and the blend function is problem dependent, and it seems difficult to give a rule valid under all circumstances. However, the choice $\omega = 0$ did not give good results in our example. In the example shown in Fig. 7 we have used a blend region determined by $\omega = 10h$. 
5. Concluding remarks

The extrapolation part of our three-step gridding method is by far the most CPU-demanding step, and the computational cost grows rapidly with the size of the problem. In order to illustrate this we have run the problem connected to Fig. 5 for different grid sizes.

To be able to compare solutions at different grid resolutions we use the discrete versions of the $L_2$ inner product and norm on $G(E)$. With $|E|$ denoting the number of elements in $E$ we define

$$ (u, v)_h = \frac{1}{|E|} \sum_{i \in E} u_i v_i $$

and

$$ \|u\|_{h,2} = (u, u)_h^{1/2}. $$

Let $x^k$ be the $k$th iterate of the conjugate gradient method for (5). The iterations were stopped when the $k$th residual $r^k = z - Q^1 x^k$ satisfied

$$ \|r^k\|_{h,2} < 10^{-8}. $$

As start vector we used $x^0 = 0$. The number of iterations is shown in Table 1. Since the set $E$, in this case, is a square, it can be shown that the spectral condition number of the operator $Q^1$ is $O(h^{-4})$. Thus, the convergence theory of the CG-method predicts that the number of iterations is $O(|E|)$, cf. [6], and it is easily seen from the table that this is indeed the case.

The number of iterations increases with decreasing grid spacing in a similar manner as the other examples presented in Section 4. The reason is that the linear systems are ill-conditioned in the sense that the spectral condition number goes to infinity as $h$ goes to zero. Since the number of conjugate gradient iterations increases proportionally to the square root of the spectral condition number, the number of iterations becomes very large as $|E|$ increases, and the extrapolation part of our scheme becomes very CPU-demanding. This property of our scheme is due to the fact that we are unable to compute a sufficiently good start vector for the CG-iteration.

In the examples we have used the null vector as the start vector. Another possibility is to use a local interpolation scheme, as in the regularization process, to compute start values at the nodes.
of $E$. We have experimented with this, but we have not been able to construct initial start vectors which reduces the number of iterations significantly.

We also remark that the extrapolation operators we have considered represent consistent approximations of partial differential operators. One example of this is given by a proper scaling of $Q^1$, which approximates the equation $u_{xxxx} + u_{yyyy} = 0$. Thus we are led to believe that fast methods for solving linear elliptic partial differential equations may constitute good methods for solving our extrapolation problem. Clearly this is the case when the set $E$ is a simple rectangle. Then the extrapolation problem is a standard boundary value problem which, for many choices of $Q$, can be solved by very fast direct methods. In the more general case when $E$ may be a union of many regions of arbitrary shape, the class of methods referred to as domain decomposition strategies may be of interest.

The last step in our three-stages procedure is the most CPU-demanding step, and it is also this step that more or less determines the geometric properties of our final approximation. Hence, further study of both equation solvers and suitable extrapolation operators is called for. One particularly interesting problem arises if we allow the operator $Q$ to be nonlinear. This would enable us to study extrapolation under very general minimization requirements, but we also would have to solve nonlinear boundary value problems.

We believe that the main advantages of the method are that it is rather general and it is easily implemented. The main disadvantages are that the approximant does not depend continuously on the data and that the method becomes time consuming as the grid spacing is reduced. The problem of lacking continuous dependence is a hard one, and we believe that it is a fairly common problem for various methods in this field, cf. the local approximation example in Section 4.1. In our three-stage method, we believe that the lack of continuity can be reduced by introducing a certain "reliability" measure for each grid point in the set $D$. The "reliability" should be large for nodes close to many data points and small when the closest data point is far away. For this purpose, further study of the use of density functions is interesting.

| $|E|$  | Iterations |
|------|------------|
| 100  | 31         |
| 400  | 109        |
| 900  | 226        |
| 1600 | 385        |
| 2500 | 580        |
| 5776 | 1287       |
| 10000| 2170       |
References


