

Recent Developments in the Theory and Application of the Sparse Grid Combination Technique

Markus Hegland, Brendan Harding, Christoph Kowitz, Dirk Pflüger, and Peter Strazdins

Abstract Substantial modifications of both the choice of the grids, the combination coefficients, the parallel data structures and the algorithms used for the combination technique lead to numerical methods which are scalable. This is demonstrated by the provision of error and complexity bounds and in performance studies based on a state of the art code for the solution of the gyrokinetic equations of plasma physics. The key ideas for a new fault-tolerant combination technique are mentioned. New algorithms for both initial- and eigenvalue problems have been developed and are shown to have good performance.

1 Introduction

The solution of moderate- to high-dimensional PDEs (larger than four dimensions) comes with a high demand for computational power. This is due to the curse of dimensionality, which manifests itself by the fact that very large computational grids are required even for moderate accuracy. In fact, the grid sizes are an exponential function of the dimension of the problem. Regular grids are thus not feasible even when future exascale systems are to be utilized. Fortunately, hierarchical

M. Hegland (✉) • B. Harding
Mathematical Sciences Institute, The Australian National University, Canberra, Australia
e-mail: markus.hegland@anu.edu.au; brendan.harding@anu.edu.au

C. Kowitz
Chair of Scientific Computing, Technische Universität München, Munich, Germany
e-mail: kowitz@in.tum.de

D. Pflüger
Institute for Parallel and Distributed Systems, University of Stuttgart, Stuttgart, Germany
e-mail: dirk.pflueger@ipvs.uni-stuttgart.de

P. Strazdins
Engineering and Computer Science, The Australian National University, Canberra, Australia
e-mail: peter.strazdins@anu.edu.au

discretization schemes come to the rescue. So-called sparse grids [53] mitigate the curse of dimensionality to a large extent.

Nonetheless, the need for HPC resources remains. The aim of two recent projects, one (EXAHD) within the German priority program “Software for exascale computing” and one supported through an Australian Linkage grant and Fujitsu Laboratories of Europe, has been to study the sparse grid combination technique for the solution of moderate-dimensional PDEs which arise in plasma physics for the simulation of hot fusion plasmas. The combination technique is well-suited for such large-scale simulations on future exascale systems, as it adds a second level of parallelism which admits scalability. Furthermore, its hierarchical principle can be used to support algorithm-based fault tolerance [38, 46]. In this work, we focus on recent developments with respect to the theory and application of the underlying methodology, the *sparse grid combination technique*.

The sparse grid combination technique utilizes numerical solutions $u(\gamma)$ of partial differential equations computed for selected values of the parameter vector γ which controls the underlying grids. As the name suggests, the method then proceeds by computing a linear combination of the component solutions $u(\gamma)$:

$$u_I = \sum_{\gamma \in I} c_\gamma u(\gamma) . \quad (1)$$

Computationally, the combination technique thus consists of a reduction operation which evaluates the linear combination of the computationally independent components $u(\gamma)$. A similar structure is commonly found in data analytic problems and is exploited by the Map Reduce method. Since the inception of the combination technique, parallel algorithms were studied which made use of the computational structure [15, 18, 19]. The current work is based on the same principles as these earlier works, see [2, 24, 25, 28, 31, 32, 38–40, 48].

The combination technique computes a sparse grid approximation without having to implement complex sparse grid data structures. The result is a proper sparse grid function. In the case of the interpolation problem one typically obtains the exact sparse grid interpolant but for other problems (like finite element solutions) one obtains an approximating sparse grid function. Mathematically, the combination technique is an extrapolation method, and the accuracy is established using error expansions, see [5, 44, 45]. However, specific error expansions are only known for simple cases. Some recent work on errors of the sparse grid combination technique can be found in [16, 20, 21, 47]. The scarcity of theoretical results, however, did not stop its popularity in applications. Examples include partial differential equations in fluid dynamics, the advection and advection-diffusion equation, the Schrödinger equation, financial mathematics, and machine learning, see, e.g., [8–13, 17, 41, 51]. However, as the combination technique is an extrapolation method, it is inherently unstable and large errors may occur if the error expansions do not hold. This is further discussed in [30] where also a stabilized approach, the so-called Opticom method, is analyzed. Several new applications based on this stabilized

approach are discussed in [1, 7, 23, 26, 35, 51, 52]. Other non-standard combination approximations are considered in [4, 35, 37, 43].

The main application considered in the following deals with the solution of the gyrokinetic equations by the software code GENE [14]. These equations are an approximation for the case of a small Larmor-radius of the Vlasov equations for densities f_s of plasmas,

$$\frac{\partial f_s}{\partial t} + \mathbf{v} \cdot \frac{\partial f_s}{\partial \mathbf{x}} + \frac{q_s}{m_s} (\mathbf{E} + \mathbf{v} \times \mathbf{B}) \cdot \frac{\partial f_s}{\partial \mathbf{v}} = 0. \quad (2)$$

The densities are distribution functions over the state space and E and B are the electrostatic and electromagnetic fields (both external and induced by the plasma), \mathbf{v} is the velocity and \mathbf{x} the location. The fields E and B are then the solution of the Maxwell equations for the charge and current densities defined by

$$\rho(\mathbf{x}, t) = \sum_s q_s \int f_s(\mathbf{x}, \mathbf{v}, t) d\mathbf{v}, \quad \text{and} \quad \mathbf{j}(\mathbf{x}, t) = \sum_s q_s \int f_s(\mathbf{x}, \mathbf{v}, t) \mathbf{v} d\mathbf{v}. \quad (3)$$

While the state space has 6 dimensions (3 space and 3 velocity), the gyrokinetic equations reduce this to 5 dimensions. The index s numbers the different species (ions and electrons). The numerical scheme uses both finite differences and spectral approximations. As complex Fourier transforms are used, the densities f_s are complex.

In Sect. 2 a general combination technique suitable for our application is discussed. In this section the set I occurring in the combination formula (1) uniquely determines the combination coefficients c_γ in that formula. Some parallel algorithms and data structures supporting the sparse grid combination technique are presented in Sect. 3. In order to stabilize the combination technique, the combination coefficients need to be modified and even chosen dependent on the solution. This is covered in Sect. 4. An important application area relates to eigenvalue problems in Sect. 5, where we cover challenges and algorithms for this problem.

2 A Class of Combination Techniques

Here we call a *combination technique* a method which is obtained by substituting some of the hierarchical surpluses by zero. This includes the traditional sparse grid combination technique [19], the truncated combination technique [4], dimension adaptive variants [10, 29] and even some of the fault tolerant methods [24]. The motivation for this larger class is that often the basic *error splitting assumption*—which can be viewed as an assumption about the surplus—does not hold in these cases. We will now formally define this combination technique.

We assume that we have at our disposition a computer code which is able to produce approximations of some real or complex number, some vector or some

function. We denote the *quantity of interest* by u and assume that the space of all possible u is a Euclidean vector space (including the numbers) or a Hilbert space of functions. The computer codes are assumed to compute a very special class of approximations $u(\gamma)$ which in some way are associated with regular d -dimensional grids with step size $h_i = 2^{-\gamma_i}$ in the i -th coordinate. For simplicity we will assume that in principle our code can compute $u(\gamma)$ for any $\gamma \in \mathbb{N}_0^d$. Furthermore, $u(\gamma) \in V(\gamma)$ where the spaces $V(\gamma) \subset V$ are hierarchical, such that $V(\alpha) \subset V(\beta)$ when $\alpha \leq \beta$ (i.e. where $\alpha_i \leq \beta_i$ for all $i = 1, \dots, d$). For example, if $V = \mathbb{R}$ then so are all $V(\gamma) = \mathbb{R}$. Another example is the space of functions with bounded (in L_2) mixed derivatives $V = H_{\text{mix}}^1([0, 1]^d)$. In this case one may choose $V(\gamma)$ to be appropriate spaces of multilinear functions.

The quantities of interest include solutions of partial differential equations, minima of convex functionals and eigenvalues and eigenfunctions of differential operators. They may also be functions or functionals of solutions of partial differential equations. They may be moments of some particle densities which themselves are solutions to some Kolmogorov, Vlasov, or Boltzmann equations. The computer codes may be based on finite difference and finite element solvers, least squares and Ritz solvers but could also just be interpolants or projections. In all these cases, the combination technique is a method which combines multiple approximations $u(\gamma)$ to get more accurate approximations. Of course the way how the underlying $u(\gamma)$ are computed will have some impact on the final combination approximation.

The combination technique is fundamentally tied to the concept of the hierarchical surplus [53] which was used to introduce the sparse grids. However, there is a subtle difference between the surplus used to define the sparse grids and the one at the foundation of the combination technique. The surplus used for sparse grids is based on the representation of functions as a series of multiples of hierarchical basis functions. In contrast, the combination technique is based on a more general decomposition. It is obtained from the following result which follows from two lemmas in chapter 4 of [22].

Proposition 1 (Hierarchical surplus) *Let $V(\gamma)$ be linear spaces with $\gamma \in \mathbb{N}_0^d$ such that $V(\alpha) \subset V(\beta)$ if $\alpha \leq \beta$ and let $u(\gamma) \in V(\gamma)$. Then there exist $w(\alpha) \in V(\alpha)$ such that*

$$\sum_{\alpha \leq \gamma} w(\alpha) = u(\gamma) . \quad (4)$$

Moreover, the $w(\gamma)$ are uniquely determined and one has

$$w(\alpha) = \sum_{\gamma \in B(\alpha)} (-1)^{|\alpha-\gamma|} u(\gamma) \quad (5)$$

where $B(\alpha) = \{\gamma \geq 0 \mid \alpha - 1 \leq \gamma \leq \alpha\}$ and $1 = (1, \dots, 1) \in \mathbb{N}^d$.

The set of γ is countable and the proposition is proved by induction over this set. Note that the equations are cumulative sums and the solution is given in the form of a finite difference. For the case of $d = 2$ and $\gamma \leq (2, 2)$ one gets the following system of equations:

$$\begin{bmatrix} u(2, 2) \\ u(1, 2) \\ u(2, 1) \\ u(0, 2) \\ u(1, 1) \\ u(2, 0) \\ u(0, 1) \\ u(1, 0) \\ u(0, 0) \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ & 1 & 1 & 1 & 1 & 1 & & & \\ & & 1 & 1 & 1 & 1 & 1 & & \\ & & & 1 & 1 & 1 & & & \\ & & & & 1 & 1 & 1 & & \\ & & & & & 1 & 1 & & \\ & & & & & & 1 & 1 & \\ & & & & & & & 1 & 1 \\ & & & & & & & & 1 \end{bmatrix} \begin{bmatrix} w(2, 2) \\ w(1, 2) \\ w(2, 1) \\ w(0, 2) \\ w(1, 1) \\ w(2, 0) \\ w(0, 1) \\ w(1, 0) \\ w(0, 0) \end{bmatrix} . \tag{6}$$

Note that all the components of the right hand side and the solution are elements of linear spaces. The vector of $w(\alpha)$ is for the example:

$$\begin{bmatrix} w(2, 2) \\ w(1, 2) \\ w(2, 1) \\ w(0, 2) \\ w(1, 1) \\ w(2, 0) \\ w(0, 1) \\ w(1, 0) \\ w(0, 0) \end{bmatrix} = \begin{bmatrix} +1 & -1 & -1 & & +1 & & & & \\ & +1 & & -1 & -1 & & +1 & & \\ & & +1 & & -1 & -1 & & +1 & \\ & & & +1 & & & -1 & & \\ & & & & +1 & & -1 & -1 & +1 \\ & & & & & +1 & & -1 & \\ & & & & & & +1 & & -1 \\ & & & & & & & +1 & -1 \\ & & & & & & & & +1 \end{bmatrix} \begin{bmatrix} u(2, 2) \\ u(1, 2) \\ u(2, 1) \\ u(0, 2) \\ u(1, 1) \\ u(2, 0) \\ u(0, 1) \\ u(1, 0) \\ u(0, 0) \end{bmatrix} . \tag{7}$$

For any set of indices $I \subset \mathbb{N}_0^d$ we now define the combination technique as any method delivering the approximation

$$u_I = \sum_{\alpha \in I} w(\alpha) . \tag{8}$$

In practice, the approximation u_I is computed directly from the $u(\gamma)$. The combination formula is directly obtained from Proposition 1 and one has

Proposition 2 *Let $u_I = \sum_{\alpha \in I} w(\alpha)$ where $w(\alpha)$ is the hierarchical surplus for the approximations $u(\gamma)$. Then there exists a subset I' of the smallest downset which contains the set I and some coefficients $c_\gamma \in \mathbb{Z}$ for $\gamma \in I'$ such that*

$$u_I = \sum_{\gamma \in I'} c_\gamma u(\gamma) . \tag{9}$$

Furthermore, one has

$$c_\gamma = \sum_{\alpha \in C(\gamma)} (-1)^{|\gamma - \alpha|} \chi_I(\alpha) \quad (10)$$

where $C(\gamma) = \{\alpha \mid \gamma \leq \alpha \leq \gamma + 1\}$ and where $\chi_I(\alpha)$ is the characteristic function of I .

The proof of this result is a direct application of Proposition 1, see also [22]. For the example $d = 2$ and $n = 2$ one gets

$$u_n^C = u(0, 2) + u(1, 1) + u(2, 0) - u(0, 1) - u(1, 0). \quad (11)$$

Note the coefficients $c_\gamma = 1$ for the finest grids, $c_\gamma = -1$ for some grids which are slightly coarser and $c_\gamma = 0$ for all the other grids. There are both positive and negative coefficients. Indeed, the results above can also be shown to be a consequence of the *inclusion-exclusion principle*. One can show that if $0 \in I$ then $\sum_{\gamma \in I} c_\gamma = 1$.

An implementation of the combination technique will thus compute a linear combination of a potentially large number of component solutions $u(\gamma)$. One thus requires two steps, first the independent computation of the components $u(\gamma)$ and then the reduction to the combination u_I . Thus the computations require a collection of computational clusters which are loosely connected. This is a great advantage on HPC systems as the need for global communication is significantly reduced to a loose coupling.

Many variants of the combination technique are obtained using the technique introduced above. They differ by their choice of the summation set I . The classical combination technique utilizes

$$I = \{\alpha \mid |\alpha| \leq n + d - 1\}. \quad (12)$$

Many variants are subsets of this set. This includes the *truncated sparse grids* [3, 4] defined by

$$I = \downarrow \{\alpha \mid |\alpha| \leq n + d - 1, \alpha \geq \beta\} \quad (13)$$

where \downarrow is the operator producing the smallest downset containing the operand. Basically the same class is considered in [49] (there called partial sparse grids):

$$I = \downarrow \{\alpha \mid |\alpha| \leq n + |\beta| - 1, \alpha \geq \beta\} \quad (14)$$

for some $\beta \geq 1$. Sparse grids with faults [24] include sets of the form

$$I = \{\alpha \mid |\alpha| \leq n + d - 1, \alpha \neq \beta\} \quad (15)$$

for some β with $|\beta| = n$. Finally, one may consider the *two-scale* combination with

$$I = \bigcup_{k=1}^d \{ \alpha \mid \alpha \leq n_0 1 + n_k e_k \} \tag{16}$$

where e_k is the standard k -th basis vector in \mathbb{R}^d . This has been considered in [3] for the case of $n_0 = n_k = n$. Another popular choice is

$$I = \{ \alpha \mid |\text{supp } \alpha| \leq k \} . \tag{17}$$

This corresponds to a truncated ANOVA-type decomposition. An alternative ANOVA decomposition is obtained by choosing $\beta^{(k)}$ with $|\text{supp } \beta^{(k)}| = k$ and setting

$$I = \bigcup_{k=1}^d \{ \alpha \mid \alpha \leq \beta^{(k)} \} . \tag{18}$$

The sets I are usually *downsets*, i.e., such that $\beta \in I$ if there exists an $\alpha \in I$ such that $\beta \leq \alpha$. Note that any downset I especially contains the zero vector. The corresponding vector space $V(I)$ typically contains the set of constant functions.

We will now consider errors. First we reconsider the error of the $u(\gamma)$. In terms of the surpluses, one has from the surplus decomposition of $u(\gamma)$ that

$$e(\gamma) = u - u(\gamma) = \sum_{\alpha \not\leq \gamma} w(\alpha) . \tag{19}$$

Let $I_s(\gamma) = \{ \alpha \mid \alpha_s > \gamma_s \}$. Then one has

$$\{ \alpha \not\leq \gamma \} = \bigcup_{s=1}^d I_s(\gamma) \tag{20}$$

as any α which is not less or equal to γ contains at least one element $\alpha_s > \gamma_s$. We now define

$$I(\gamma; \sigma) = \bigcap_{s \in \sigma} I_s(\gamma) \tag{21}$$

for any non-empty subset $\sigma \subseteq \{1, \dots, d\}$. A direct application of the inclusion-exclusion principle then leads to the *error splitting*

$$e(\gamma) = \sum_{\emptyset \neq \sigma \subseteq \{1, \dots, d\}} (-1)^{|\sigma|-1} z(\gamma, \sigma) \tag{22}$$

where

$$z(\gamma, \sigma) = \sum_{\alpha \in I(\gamma; \sigma)} w(\alpha) . \quad (23)$$

This is an ANOVA decomposition of the approximation error of $u(\gamma)$. From this one gets the result

Proposition 3 *Let $u_I = \sum_{\gamma \in I'} c_\gamma u(\gamma)$ and the combination coefficients c_γ be such that $\sum_{\gamma \in I} c_\gamma = 1$. Then*

$$u - u_I = \sum_{\emptyset \neq \sigma \subseteq \{1, \dots, d\}} (-1)^{|\sigma|-1} \sum_{\gamma \in I'} c_\gamma z(\gamma, \sigma) . \quad (24)$$

Proof This follows from the discussion above and because $0 \in I$ one has

$$u - u_I = \sum_{\gamma \in I'} c_\gamma e(\gamma) . \quad (25)$$

□

An important point to note here is that this error formula does hold for any coefficients c_γ , not just the ones defined by the general combination technique. This thus leads to a different way to choose the combination coefficients which results in a small error. We will further discuss such choices in the next section. Note that for the general combination technique the coefficients are uniquely determined by the set I . In this case one has a complete description of the error using the hierarchical surplus

$$e_I = \sum_{\alpha \notin I} w(\alpha) . \quad (26)$$

In summary, we have now two strategies to design a combination approximation: one may choose either

- the set I which contains all the $w(\alpha)$ which are larger than some threshold
- or the combination coefficients such that the sums $\sum_{\alpha \in I(\gamma, \sigma)} c_\gamma z(\gamma, \sigma)$ are small.

One approach is to select the $w(\alpha)$ adaptively, based on their size so that

$$I = \{\alpha \mid \|w(\alpha)\| \geq \epsilon\} . \quad (27)$$

Such an approach is sometimes called dimension adaptive to distinguish it from the spatially adaptive approach where grids are refined locally. One may be interested in finding an approximation for some $u(\gamma)$, for example, for $\gamma = (n, \dots, n)$. In this

case one considers

$$I = \{\alpha \leq \gamma \mid \|w(\alpha)\| \geq \epsilon\} \quad (28)$$

and one has the following error bound:

Proposition 4 *Let $I = \{\alpha \leq \gamma \mid \|w(\alpha)\| \geq \epsilon\}$ and $u(\gamma) - u_I$ be the error of the combination approximation based on the set I relative to $u(\gamma)$. Then one has the bound*

$$\|u(\gamma) - u_I\| \leq \prod_{i=1}^d (\gamma_i + 1) \epsilon . \quad (29)$$

The result is a simple application of the triangle inequality and the fact that

$$|I| = \prod_{i=1}^d (\gamma_i + 1) . \quad (30)$$

In particular, one has if all $\gamma_i = n$:

$$\|u(\gamma) - u_I\| \leq (n + 1)^d \epsilon . \quad (31)$$

While this bound is very simple, it is asymptotically (in n and d) tight due to the concentration of measure. Note also, that a similar bound for the spatially adaptive method is not available. An important point to note is that this error bound holds always, independently of how good the surplus is at approximating the exact result. For $\gamma = (n, \dots, n)$ one can combine the estimate of Proposition 4 with a bound on $u - u(\gamma)$ to obtain

$$\|u - u_I\| \leq \|u - u(\gamma)\| + \|u(\gamma) - u_I\| \leq K 4^{-n} + (n + 1)^d \epsilon . \quad (32)$$

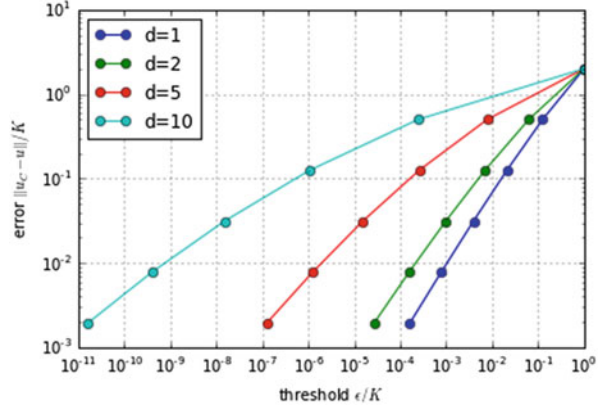
One can then choose n which minimizes this for a given ϵ by balancing the two terms. Conversely, for a given n the corresponding ϵ is given by $\epsilon_n = (n + 1)^{-d} K 4^{-n}$. In Fig. 1 we plot ϵ_n/K against $\|u - u_I\|/K$ for several different d to demonstrate how the error changes with the threshold.

While the combination approximation is the sum of the surpluses $w(\alpha)$ over all $\alpha \in I$, the result only depends on a small number of $u(\gamma)$ close to the maximal elements of I . In particular, any errors of the values $u(\alpha)$ for small α have no effect for approximations based on larger α . Thus when doing an adaptive approximation, the earlier errors are forgotten.

Finally, if one has a model for the hierarchical surplus, for example, if it is of the form

$$\|w(\alpha)\| \leq 4^{-|\alpha|} y(\alpha) \quad (33)$$

Fig. 1 Scaled error against threshold



for some bounded $y(\alpha)$ then one can get specific error bounds for the combination technique, in particular the well-known bounds for the classical sparse grid technique. In this case one gets $\|w(\alpha)\| \leq K4^{-|\alpha|}$ if one chooses $|\alpha| \geq n$ as for the classical combination technique. One can show that the terms in the error formula for the components $u(\gamma)$ satisfy

$$\|z(\gamma, \sigma)\| \leq \left(\frac{4}{3}\right)^d 4^{-\sum_{s=1}^{|\sigma|} (\gamma_{\sigma_s} + 1)} K. \tag{34}$$

3 Algorithms and Data Structures

In this section we consider the parallel implementation of the combination technique for partial differential equation solvers. For large-scale simulations, for example as being the final target for the EXAHD project in the second phase, even a single component grid (together with the data structures to solve the underlying PDE on it) will not fit into the memory of a single node any more. Furthermore, the storage of a full grid representation of a sparse grid will exceed the predicted RAM of a whole exascale machine. Furthermore, the communication overhead across a whole HPC systems' network cannot be neglected. In this section we will assume that the component grids $u(\gamma)$ are implemented as distributed regular grids. In a first stage we consider the case where the combined solution u_I is also a distributed regular grid. Later we will then discuss distributed sparse grid data structures.

The combination technique is a reduction operation combining the components according to Eq. (1). This reduction is based on the sum $u' \leftarrow u' + cu$ of a component u (we omit the parameters γ for simplicity) to the resulting combination u' (or u_I). Assume that the u and u' are distributed over P and P' processors, respectively.

The *direct SGCT algorithm* involves for each of the component processes sending all its points of u to the respective combination process. This is denoted as the

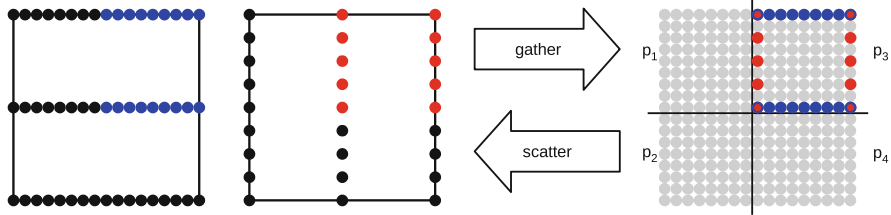


Fig. 2 Gather and scatter steps

gather stage. In a second stage, the combination processes then first *interpolates* the gathered points to the combination grid u before adding them. In a third stage, the *scatter stage*, the data on each combination process is sampled and the samples sent to the corresponding component processes, see Fig. 2.

In the direct SGCT algorithm, the components and combination are represented by the function values on the grid points or coefficients of the nodal basis. We have also considered a *hierarchical SGCT algorithm* which is based on the coefficients of the hierarchical basis which leads to a hierarchical surplus representation. When the direct SGCT algorithm is applied to these hierarchical surpluses there is no need for interpolation, and the sizes of the corresponding surplus vectors are exactly the same for both the components and the combination. However, for performance, it is necessary to *coalesce* the combination of surpluses as described in [49]. As the largest surpluses only occur for one component they do not need to be communicated. Despite the savings in the hierarchical algorithm, we found that the direct algorithm is always faster than the hierarchical, and it scales better with both n , d and the number of processes (cores). This does however require that the representation of the combined grid u' is sparse, as is described below. We also found that the formation of the hierarchical surpluses (and its inverse) took a relatively small amount of time, and concluded that, even when the data is originally stored in hierarchical form, it is faster to dehierarchize it, apply the direct algorithm and hierarchize it again [49].

New adapted algorithms and implementations have been developed with optimal communication overhead, see Fig. 3 (left) and the corresponding paper in this proceedings [27]. The gather–scatter steps described above have to be invoked multiple times for the solution of time-dependent PDEs. (We found that for eigenvalue problems it is often sufficient to call the gather–scatter only once, see the Sect. 5.) In any case, the gather–scatter step is the only remaining global communication of the combination technique and thus has to be examined well. In previous work [31] we have thus analyzed communication schemes required for the combination step in the framework of BSP-models and developed new algorithmic variants with communication that is optimal up to constant factors. This way, the overall makespan volume, the maximal communicated volume, can be drastically reduced with a slightly increased number of messages that have to be sent.

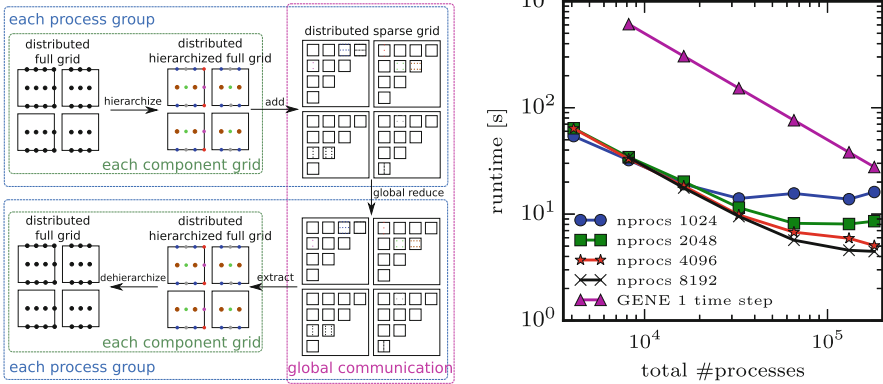


Fig. 3 Distributed hierarchical combination with optimal communication overhead (*left*) and runtime results on Hazel Hen (*right*) for different sizes of process groups (local parallelism with n_{procs} processors). The results measure only the communication (local + global) and distributed hierarchization, not the computation. The saturation for large numbers of component grids is due to the logarithmic scaling of the global reduce step for large numbers of process groups and up to 180,224 processors in total. In comparison, the time for a single time step with GENE for a process group size of 4096 is shown, see [27] in this proceedings for further details

A distributed sparse grid data structure is described in [49]. The index set I for this case is a variant of a truncated sparse grid set, see Eq. (14). Recall that the sparse grid points are obtained by taking the union of all the component grid points. As the number of sparse grid points is much less than the number of full grid points it makes sense to compute only the combinations for the sparse grid points. A sparse grid data structure has been developed which is similar to the CSR data structure used for sparse matrices. In this case one stores both information about the value u at the grid point and the location of the grid point. Due to the regularity of the sparse grid this can be done efficiently.

With optimal communication, distributed data structures and corresponding algorithms, excellent scaling can be obtained for large numbers of process groups as shown in Fig. 3 (right) on Hazel Hen, which includes local algorithmic work to hierarchize, local communication and global communication. See the corresponding paper in this proceedings [27].

4 Modified Combination Coefficients

Here we consider approximations which are based on a vector $(u(\gamma))_{\gamma \in I}$ of numerical results. It has been seen, however, that the standard way to choose the combination coefficients is not optimal and may lead to large errors. In fact one may interpret the truncated combination technique as a variant where some of the

coefficients have been chosen to be zero and the rest adapted. In the following we provide a more radical approach to choosing the coefficients c_γ . An advantage of this approach is that it does not depend so much on properties of the index set I , in fact, this set does not even need to be a downset.

A first method was considered in [30, 52] for convex optimization problems. Here, let the component approximations be

$$u(\gamma) = \operatorname{argmin}\{J(v) \mid v \in V(\gamma)\} . \tag{35}$$

Then the Opticom method, a Ritz approximation over the span of given $u(\gamma)$ computes

$$u^O = \operatorname{argmin} \left\{ J(v) \mid v = \sum_{\gamma \in I} c_\gamma u(\gamma) \right\} . \tag{36}$$

Computationally the Opticom method consists of the determination of minimization of a convex function $P(c)$ of $|I|$ variables of the form

$$\Phi(c) = J \left(\sum_{\gamma \in I} c_\gamma u(\gamma) \right) \tag{37}$$

to get the combination coefficients. Once they have been determined, the approximation u^O is then computed as in the Sects. 2 and 3. By design, one has $J(u^O) \leq J(u(\gamma))$ for all $\gamma \in I$. If I gives rise to a combination approximation u^C then one also has $J(u^O) \leq J(u^C)$. A whole family of other convex functions $\Phi(c)$ for the combination coefficients were considered in [30]. Using properties of the Bregman divergence, one can derive error bounds and quasi-optimality criteria for the Opticom method, see [52].

A similar approach was suggested for the determination of combination coefficients for faulty sets I . Let I be any set and I' be the smallest downset which contains I . Then let the $w(\alpha)$ be the surpluses computed from the set of all $u(\gamma)$ for $\gamma \in I$ and $\alpha \in I'$. Finally, let the regular combination technique be defined as

$$u^R = \sum_{\alpha \in I'} w(\alpha) \tag{38}$$

and let for any c_γ a combination technique be

$$u^C = \sum_{\gamma \in I} c_\gamma u(\gamma) . \tag{39}$$

Then the difference between the new combination technique and the regular combination technique is

$$\begin{aligned}
 u^C - u^R &= \sum_{\gamma \in I} c_\gamma u(\gamma) - \sum_{\alpha \in I'} w(\alpha) \\
 &= \sum_{\gamma \in I} c_\gamma \sum_{\alpha \leq \gamma} w(\alpha) - \sum_{\alpha \in I'} w(\alpha) \\
 &= \sum_{\alpha \in I'} w(\alpha) \left(\sum_{\gamma \in I(\alpha)} c_\gamma - 1 \right)
 \end{aligned} \tag{40}$$

where $I(\alpha) = \{\gamma \in I \mid \gamma \geq \alpha\}$. Using the triangle inequality one obtains

$$\|u^C - u^R\| \leq \Phi(c) \tag{41}$$

with

$$\Phi(c) = \sum_{\alpha \in I'} \theta(\alpha) \left| \sum_{\gamma \in I(\alpha)} c_\gamma - 1 \right|, \tag{42}$$

where θ is such that $\|w(\alpha)\| \leq \theta(\alpha)$. Minimizing the $\Phi(c)$ thus seems to lead to a good choice of combination coefficients, and this is confirmed by experiments as well [22]. The resulting combination technique forms the basis for a new fault-tolerant approach which has been discussed in [24].

5 Computing Eigenvalues and Eigenvectors

Here we consider the eigenvalue problem in V where one would like to compute complex eigenvalues λ and the corresponding eigenvectors u such that

$$Lu = \lambda u, \tag{43}$$

where L is a given linear operator defined on V . We assume we have a code which computes approximations $\lambda(\gamma) \in \mathbb{C}$ and $u_\lambda(\gamma) \in V(\gamma)$ of the eigenvalues λ and the corresponding eigenvectors u . We have chosen to discuss the eigenvalue problem separately as it does exhibit particular challenges which do not appear for initial and boundary value problems.

Consider now the determination of the *eigenvalues* λ . Note that one typically has a large number of eigenvalues for any given operator L . First one needs to decide which eigenvalue to compute. For example, if one is interested in stability of a system, one would like to determine the eigenvalue with the largest real part. It is

possible to use the general combination technique, however, one needs to make sure that the (non-zero) combination coefficients c_γ used are such that the eigenvectors of $L(\gamma)$ contain approximations of the eigenvector u which is of interest. However, computing the surplus $v(\alpha)$ for the eigenvalues $\lambda(\gamma)$ and including all the ones which satisfy $|v(\alpha)| \geq \epsilon$ for some ϵ would be a good way to make sure that we get a good result. Furthermore, the error bound given in Sect. 2 does hold here. As any surplus $v(\alpha)$ does only depend on the values $\lambda(\gamma)$ for γ close to α any earlier $\lambda(\gamma)$ with a large error will not influence the final result. Practical computations confirmed the effectiveness of this approach, see [34]. If one knows which spaces $V(\gamma)$ produce reasonable approximations for the eigenvector corresponding to some eigenvalue λ then one can define a set $I(\lambda)$ containing only those γ . Combinations over $I(\gamma)$ will then provide good approximations of λ . (However, as stated above, the combination technique is asymptotically stable against wrong or non-existing eigenvectors.)

Computing the eigenvectors faces the same problem one has for computing the eigenvalues. In addition, however, one has an extra challenge as the eigenvectors are only determined up to some complex factor. In particular, if one uses the eigenvectors $u(\gamma)$ to compute the surplus functions $w(\alpha)$ one may get very wrong results. One way to deal with this is to first normalize the eigenvectors. For this one needs a functional $s \in V^*$. One then replaces the $u(\gamma)$ by $u(\gamma)/\langle s, u(\gamma) \rangle$ when computing the surplus, i.e., one solves the surplus equations

$$\sum_{\alpha \leq \gamma} w(\alpha) = \frac{u(\gamma)}{\langle s, u(\gamma) \rangle} \tag{44}$$

and computes the combination approximation as

$$u_I = \sum_{\gamma \in I} c_\gamma \frac{u(\gamma)}{\langle s, u(\gamma) \rangle} . \tag{45}$$

In practice, this did give good results and it appears reasonable that bounds on the so computed surplus provide a foundation for the error analysis. In any case the error bound of the adaptive method holds. Actually, this bound even holds when the eigenvectors are not normalized. The advantage of the normalization is really that the number of surpluses to include are much smaller—i.e. a computational advantage. Practical experiments also confirmed this. It remains to be shown that error splitting assumptions are typically invariant under the scaling done above.

5.1 An Opticom Approach for Solving the Eigenvalue Problem

An approach to solving the eigenvalue problem which does not require scaling has been proposed and investigated by Kowitz and collaborators [34, 36]. The approach

is based on a minimization problem which determines combination coefficients in a similar manner as the opticom method in Sect. 3. It is assumed that I is given and the $u(\gamma)$ for $\gamma \in I$ have been computed and solve $L(\gamma)u(\gamma) = \lambda(\gamma)u(\gamma)$. Let the matrix $G = [u(\gamma)]_{\gamma \in I}$ and the vector $c = [c_\gamma]_{\gamma \in I}^T$, then the combination approximation for the eigenvector can be written as the matrix-vector product

$$Gc = \sum_{\gamma \in I} c_\gamma u(\gamma). \quad (46)$$

This eigenvalue problem can be solved by computing

$$(c, \lambda) = \operatorname{argmin}_{c, \lambda} \|LGc - \lambda Gc\| \quad (47)$$

with the normal equations

$$(LG - \lambda G)^*(LG - \lambda G)c = 0 \quad (48)$$

for the solution of c . Osborne et al. [33, 42] solved this by considering the problem

$$\begin{pmatrix} K(\lambda) & t \\ s^* & 0 \end{pmatrix} \begin{pmatrix} c \\ \beta \end{pmatrix} = \begin{pmatrix} 0 \\ 1 \end{pmatrix} \quad (49)$$

with $K(\lambda) = (LG - \lambda G)^*(LG - \lambda G)$. Here λ is a parameter. One obtains the solution

$$\beta(\lambda) = -\langle s^*, K(\lambda)^{-1}t \rangle^{-1} \quad (50)$$

for which one then uses Newton's method to solve $\beta(\lambda) = 0$ with respect to λ . With $\beta(\lambda) = 0$ it follows that $K(\lambda)c = 0$ and $\langle s^*, c \rangle = 1$. Thus one obtains a normalized solution of the nonlinear eigenvalue problem (i.e., where λ occurs in a nonlinear way in $K(\lambda)$).

Another approach for obtaining the least squares solution is its interpretation as an overdetermined eigenvalue problem. Das et al. [6] developed an algorithm based on the QZ decomposition which allows the computation of the eigenvalue and the eigenvector in $\mathcal{O}(mn)$ complexity, where $n = |I|$ and $m = |V|$.

The approaches have both been investigated for a simple test problem (see left of Fig. 4) and for large eigenvalue computations with GENE (see right of Fig. 4). The combination approximations (though computed serially here) can be usually obtained faster than the full grid approximations. Note that the run-times here have been obtained in a prototypical implementation before the development of the scalable algorithms described in Sect. 3. For large problems, the combination approximation can be expected to be even significantly faster as the combination technique exhibits a better parallel scalability than the full grid solution. For further details, see [34, 36].

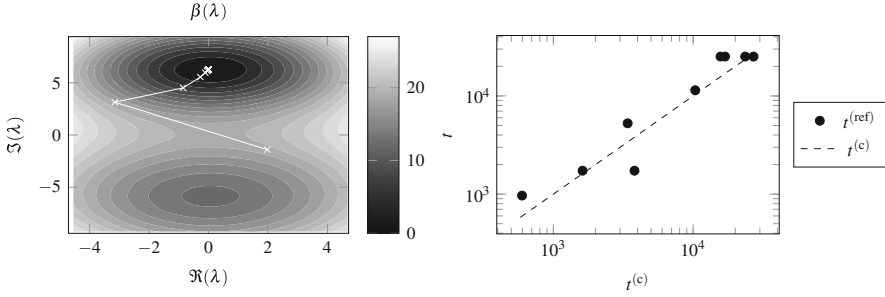


Fig. 4 The convergence of the Newton iteration towards the root of $\beta(\lambda)$ for the simple test problem (*left*) and the time for obtaining the combination approximation $i^{(c)}$ compared to the time to compute an eigenpair on a full grid of similar accuracy $i^{(\text{ref})}$ for linear GENE computations (*right*)

5.2 Iterative Refinement and Iterative Methods

Besides the adaptation of the combination coefficients, the combination technique for eigenvalue problems can also be improved by refining the $u(\gamma)$ iteratively. Based on the iterative refinement procedure introduced by Wilkinson [50], the approximation of the eigenvalue λ_I and the corresponding eigenvector u_I can be improved towards λ and u with corrections $\Delta\lambda$ and Δu by

$$u = u_I + \Delta u \quad \lambda = \lambda_I + \Delta\lambda . \tag{51}$$

Putting this into $0 = Lu - \lambda u$, the corrections can be obtained by solving

$$0 = Lu_I - \lambda_I u_I - \Delta\lambda u_I + L\Delta u - \lambda_I \Delta u, \tag{52}$$

where the quadratic term $\Delta\lambda \Delta u$ is neglected. This system is underdetermined. An additional scaling condition $\langle s^*, \Delta u \rangle = 0$ with $s \in V$ ensures that the correction Δu does not change the magnitude of u_I . Solving the linear system

$$\begin{pmatrix} L - \lambda_I I & u_I \\ s^* & 0 \end{pmatrix} \begin{pmatrix} \Delta u \\ \Delta\lambda \end{pmatrix} = \begin{pmatrix} \lambda_I u_I - Lu_I \\ 0 \end{pmatrix}, \tag{53}$$

we obtain the corrections $\Delta\lambda$ and Δu . The linear operator L has a large rank and its inversion is generally infeasible for high-dimensional settings. Nevertheless computing a single matrix vector product Lu_I is feasible, so that the right-hand side is easily computed. In the framework of the combination technique the corrections Δu and $\Delta\lambda$ are computed on each subspace $V(\gamma)$. Therefore, the residual $r = Lu - \lambda u$ and the initial combination approximation u_I are projected on $V(\gamma)$ using suitable prolongation operators [18]. The corrections $\Delta u(\gamma)$ and $\Delta\lambda(\gamma)$ are

computed on each subspace $V(\gamma)$ by solving

$$\begin{pmatrix} L(\gamma) - \lambda_I I & u_I(\gamma) \\ s^*(\gamma) & 0 \end{pmatrix} \begin{pmatrix} \Delta u(\gamma) \\ \Delta \lambda(\gamma) \end{pmatrix} = \begin{pmatrix} -r(\gamma) \\ 0 \end{pmatrix}. \quad (54)$$

Here, the significantly smaller rank of $L(\gamma)$ allows the solution of the linear system with feasible effort. The corrections from each subspace $V(\gamma)$ are then combined using the standard combination coefficients c_γ by

$$\Delta u_I = \sum_{\gamma \in I} c_\gamma \Delta u_I(\gamma) \quad \Delta \lambda_I = \sum_{\gamma \in I} c_\gamma \Delta \lambda_I(\gamma). \quad (55)$$

After adding the correction to u_I and λ_I , the process can be repeated up to marginal $\Delta \lambda_I$ and Δu_I .

Instead of using the standard combination coefficients c_γ , we can also adapt the combination coefficients in order to minimize the residual r . The minimizer

$$(\Delta u, \Delta \lambda) = \operatorname{argmin}_c \|r - \lambda_I u_I + L \Delta u_I - \lambda_I \Delta u_I - \Delta \lambda_I u_I\| \quad (56)$$

is then the best combination of the corrections. Both approaches have been tested for the Poisson problem as well as GENE simulations. For details see [34].

6 Conclusions

Early work on the combination technique revealed that it leads to a suitable method for the solution of simple boundary value problems on computing clusters. The work presented here demonstrated, that if combined with strongly scalable solvers for the components, one can develop an approach which is suitable for exascale architectures. This was investigated for the plasma physics code GENE which was used to solve initial and eigenvalue problems and stationary solutions. In addition to the 2 levels of parallelism exhibited by the combination technique, the flexibility of the choice of the combination coefficients led to a totally new approach to algorithm-based fault tolerance which further enhanced the scalability of the approach.

Acknowledgements The work presented here reviews some results of a German-Australian collaboration going over several years which was supported by grants from the German DFG (SPP-1648 SPPEXA: EXAHD) and the Australian ARC (LP110200410), contributions by Fujitsu Laboratories of Europe (FLE) and involved researchers from the ANU, FLE, TUM, and the Universities of Stuttgart and Bonn. Contributors to this research included Stephen Roberts, Jay Larson, Moshin Ali, Ross Nobes, James Southern, Nick Wilson, Hans-Joachim Bungartz, Valeriy Khakhutskyy, Alfredo Hinojosa, Mario Heene, Michael Griebel, Jochen Garcke, Rico Jacob, Philip Hupp, Yuan Fang, Matthias Wong, Vivien Challis and several others.

References

1. Ali, M.M., Southern, J., Strazdins, P.E., Harding, B.: Application level fault recovery: Using fault-tolerant open MPI in a PDE solver. In: 2014 IEEE International Parallel & Distributed Processing Symposium Workshops, Phoenix, 19–23 May 2014, pp. 1169–1178. IEEE (2014)
2. Ali, M.M., Strazdins, P.E., Harding, B., Hegland, M., Larson, J.W.: A fault-tolerant gyrokinetic plasma application using the sparse grid combination technique. In: Proceedings of the 2015 International Conference on High Performance Computing & Simulation (HPCS 2015), pp. 499–507. IEEE, Amsterdam (2015). Outstanding paper award
3. Benk, J., Bungartz, H.J., Nagy, A.E., Schraufstetter, S.: Variants of the combination technique for multi-dimensional option pricing. In: Günther, M., Bartel, A., Brunk, M., Schöps, S., Striebel, M. (eds.) Progress in Industrial Mathematics at ECMI 2010, pp. 231–237. Springer, Berlin/Heidelberg (2010)
4. Benk, J., Pflüger, D.: Hybrid parallel solutions of the Black-Scholes PDE with the truncated combination technique. In: Smari, W.W., Zeljkovic, V. (eds.) 2012 International Conference on High Performance Computing & Simulation, HPCS 2012, Madrid, 2–6 July 2012, pp. 678–683. IEEE (2012)
5. Bungartz, H.J., Griebel, M., Rüde, U.: Extrapolation, combination, and sparse grid techniques for elliptic boundary value problems. *Comput. Method. Appl. M.* **116**(1–4), 243–252 (1994)
6. Das, S., Neumaier, A.: Solving overdetermined eigenvalue problems. *SIAM J. Sci. Comput.* **35**(2), 541–560 (2013)
7. Fang, Y.: One dimensional combination technique and its implementation. *ANZIAM J. Electron. Suppl.* **52**(C), C644–C660 (2010)
8. Franz, S., Liu, F., Roos, H.G., Stynes, M., Zhou, A.: The combination technique for a two-dimensional convection-diffusion problem with exponential layers. *Appl. Math.* **54**(3), 203–223 (2009)
9. Garcke, J.: Regression with the optimised combination technique. In: Proceedings of the 23rd International Conference on Machine Learning (ICML 2006), vol. 2006, pp. 321–328. ACM, New York (2006)
10. Garcke, J.: A dimension adaptive sparse grid combination technique for machine learning. *ANZIAM J.* **48**(C), C725–C740 (2007)
11. Garcke, J., Griebel, M.: On the computation of the eigenproblems of hydrogen and helium in strong magnetic and electric fields with the sparse grid combination technique. *J. Comput. Phys.* **165**(2), 694–716 (2000)
12. Garcke, J., Hegland, M.: Fitting multidimensional data using gradient penalties and combination techniques. In: Modeling, Simulation and Optimization of Complex Processes, pp. 235–248. Springer, Berlin (2008)
13. Garcke, J., Hegland, M.: Fitting multidimensional data using gradient penalties and the sparse grid combination technique. *Computing* **84**(1–2), 1–25 (2009)
14. Gene Development Team: GENE. <http://www.genecode.org/>
15. Griebel, M.: The combination technique for the sparse grid solution of PDEs on multiprocessor machines. *Parallel Process. Lett.* **2**, 61–70 (1992)
16. Griebel, M., Harbrecht, H.: On the convergence of the combination technique. *Lect. Notes Comput. Sci.* **97**, 55–74 (2014)
17. Griebel, M., Thurner, V.: The efficient solution of fluid dynamics problems by the combination technique. *Int. J. Numer. Method. H.* **5**(3), 251–269 (1995)
18. Griebel, M., Huber, W., Rüde, U., Störckuhl, T.: The combination technique for parallel sparse-grid-preconditioning or -solution of PDEs on workstation networks. In: Bougé, L., Cosnard, M., Robert, Y., Trystram, D. (eds.) Parallel Processing: CONPAR 92 – VAPP V, Lecture Notes in Computer Science. vol. 634, pp. 217–228. Springer, Berlin/Heidelberg/London (1992). Proceedings of the Second Joint International Conference on Vector and Parallel Processing, Lyon, 1–4 Sept 1992

19. Griebel, M., Schneider, M., Zenger, C.: A combination technique for the solution of sparse grid problems. In: *Iterative Methods in Linear Algebra* (Brussels, 1991), pp. 263–281. North-Holland, Amsterdam (1992)
20. Harding, B.: Adaptive sparse grids and extrapolation techniques. In: *Proceedings of Sparse Grids and Applications 2014*. Lecture Notes in Computational Science and Engineering, vol. 109, pp. 79–102. Springer, New York (2015)
21. Harding, B.: Combination technique coefficients via error splittings. *ANZIAM J.* **56**, C355–C368 (2016). (Online)
22. Harding, B.: Fault tolerant computation of hyperbolic PDEs with the sparse grid combination technique. Ph.D. thesis, The Australian National University (2016)
23. Harding, B., Hegland, M.: A robust combination technique. *ANZIAM J. Electron. Suppl.* **54(C)**, C394–C411 (2012)
24. Harding, B., Hegland, M.: A parallel fault tolerant combination technique. *Adv. Parallel Comput.* **25**, 584–592 (2014)
25. Harding, B., Hegland, M.: Robust solutions to PDEs with multiple grids. In: Garcke, J., Pflüger, D. (eds.) *Sparse Grids and Applications*, Munich 2012. Lecture Notes in Computer Science, vol. 97, pp. 171–193. Springer, Cham (2014)
26. Harding, B., Hegland, M., Larson, J.W., Southern, J.: Fault tolerant computation with the sparse grid combination technique. *SIAM J. Sci. Comput.* **37**(3), C331–C353 (2015)
27. Heene, M., Pflüger, D.: Scalable algorithms for the solution of higher-dimensional PDEs. In: *Proceedings of SPPEXA Symposium 2016*. Lecture Notes in Computational Science and Engineering. Springer, Berlin/Heidelberg (2016)
28. Heene, M., Kowitz, C., Pflüger, D.: Load balancing for massively parallel computations with the sparse grid combination technique. In: Bader, M., Bungartz, H.J., Bode, A., Gerndt, M., Joubert, G.R. (eds.) *Parallel Computing: Accelerating Computational Science and Engineering (CSE)*, pp. 574–583. IOS Press, Amsterdam (2014)
29. Hegland, M.: Adaptive sparse grids. In: Burrage, K., Sidje, R.B. (eds.) *Proceedings of 10th Computational Techniques and Applications Conference CTAC-2001*, vol. 44, pp. C335–C353 (2003)
30. Hegland, M., Garcke, J., Challis, V.: The combination technique and some generalisations. *Linear Algebra Appl.* **420**(2–3), 249–275 (2007)
31. Hupp, P., Jacob, R., Heene, M., Pflüger, D., Hegland, M.: Global communication schemes for the sparse grid combination technique. *Adv. Parallel Comput.* **25**, 564–573 (2014)
32. Hupp, P., Heene, M., Jacob, R., Pflüger, D.: Global communication schemes for the numerical solution of high-dimensional PDEs. *Parallel Comput.* **52**, 78–105 (2016)
33. Jennings, L.S., Osborne, M.: Generalized eigenvalue problems for rectangular matrices. *IMA J. Appl. Math.* **20**(4), 443–458 (1977)
34. Kowitz, C.: Applying the sparse grid combination technique in Linear Gyrokinetics. Ph.D. thesis, Technische Universität München (2016)
35. Kowitz, C., Hegland, M.: The sparse grid combination technique for computing eigenvalues in linear gyrokinetics. *Procedia Comput. Sci.* **18**, 449–458 (2013). 2013 International Conference on Computational Science
36. Kowitz, C., Hegland, M.: An opticom method for computing eigenpairs. In: Garcke, J., Pflüger, D. (eds.) *Sparse Grids and Applications*, Munich 2012 SE – 10. Lecture Notes in Computer Science, vol. 97, pp. 239–253. Springer, Cham (2014)
37. Kowitz, C., Pflüger, D., Jenko, F., Hegland, M.: The combination technique for the initial value problem in linear gyrokinetics. *Lecture Notes in Computer Science*, vol. 88, pp. 205–222. Springer, Heidelberg (2013)
38. Larson, J.W., Hegland, M., Harding, B., Roberts, S., Stals, L., Rendell, A., Strazdins, P., Ali, M.M., Kowitz, C., Nobes, R., Southern, J., Wilson, N., Li, M., Oishi, Y.: Fault-tolerant grid-based solvers: Combining concepts from sparse grids and mapreduce. *Procedia Comput. Sci.* **18**, 130–139 (2013). 2013 International Conference on Computational Science
39. Larson, J.W., Strazdins, P.E., Hegland, M., Harding, B., Roberts, S.G., Stals, L., Rendell, A.P., Ali, M.M., Southern, J.: Managing complexity in the parallel sparse grid combination

- technique. In: Bader, M., Bode, A., Bungartz, H.J., Gerndt, M., Joubert, G.R., Peters, F.J. (eds.) PARCO. Advances in Parallel Computing, vol. 25, pp. 593–602. IOS Press, Amsterdam (2013)
40. Larson, J., Strazdins, P., Hegland, M., Harding, B., Roberts, S., Stals, L., Rendell, A., Ali, M., Southern, J.: Managing complexity in the parallel sparse grid combination technique. *Adv. Parallel Comput.* **25**, 593–602 (2014)
 41. Lastdrager, B., Koren, B., Verwer, J.: The sparse-grid combination technique applied to time-dependent advection problems. In: *Multigrid Methods, VI (Gent, 1999)*. Lecture Notes of Computer Science & Engineering, vol. 14, pp. 143–149. Springer, Berlin (2000)
 42. Osborne, M.R.: A new method for the solution of eigenvalue problems. *Comput. J.* **7**(3), 228–232 (1964)
 43. Parra Hinojosa, A., Kowitz, C., Heene, M., Pflüger, D., Bungartz, H.J.: Towards a fault-tolerant, scalable implementation of GENE. In: *Recent Trends in Computation Engineering – CE2014*. Lecture Notes in Computer Science, vol. 105, pp. 47–65. Springer, Cham (2015)
 44. Pflaum, C.: Convergence of the combination technique for second-order elliptic differential equations. *SIAM J. Numer. Anal.* **34**(6), 2431–2455 (1997)
 45. Pflaum, C., Zhou, A.: Error analysis of the combination technique. *Numer. Math.* **84**(2), 327–350 (1999)
 46. Pflüger, D., Bungartz, H.J., Griebel, M., Jenko, F., Dannert, T., Heene, M., Kowitz, C., Parra Hinojosa, A., Zaspel, P.: EXAHD: An exa-scalable two-level sparse grid approach for higher-dimensional problems in plasma physics and beyond. In: *Euro-Par 2014: Parallel Processing Workshops, Porto*. Lecture Notes in Computer Science, vol. 8806, pp. 565–576. Springer, Cham (2014)
 47. Reisinger, C.: Analysis of linear difference schemes in the sparse grid combination technique. *IMA J. Numer. Anal.* **33**(2), 544–581 (2013)
 48. Strazdins, P.E., Ali, M.M., Harding, B.: Highly scalable algorithms for the sparse grid combination technique. In: *IPDPS Workshops, Hyderabad*, pp. 941–950. IEEE (2015)
 49. Strazdins, P.E., Ali, M.M., Harding, B.: The design and analysis of two highly scalable sparse grid combination algorithms (2015, under review)
 50. Wilkinson, J.H.: *Rounding Errors in Algebraic Processes*. Her Majesty’s Stationery Office, London (1963)
 51. Wong, M., Hegland, M.: Maximum a posteriori density estimation and the sparse grid combination technique. *ANZIAM J. Electron. Suppl.* **54**(C), C508–C522 (2012)
 52. Wong, M., Hegland, M.: Opticom and the iterative combination technique for convex minimisation. *Lect. Notes Comput. Sci.* **97**, 317–336 (2014)
 53. Zenger, C.: Sparse grids. In: Hackbusch, W. (ed.) *Parallel Algorithms for Partial Differential Equations*. Notes on Numerical Fluid Mechanics, vol. 31, pp. 241–251. Vieweg, Braunschweig (1991)