

Max-Planck-Institut  
für Mathematik  
in den Naturwissenschaften  
Leipzig

Solution of linear systems in high spatial  
dimensions

by

*Wolfgang Hackbusch*

Preprint no.: 5

2015





# Solution of Linear Systems in High Spatial Dimensions

Wolfgang Hackbusch

Max-Planck-Institut für *Mathematik in den Naturwissenschaften*,  
Inselstr. 22-26, 04103 Leipzig, Germany  
wh@mis.mpg.de

## Abstract

We give an overview of various methods based on tensor structured techniques for the solution of linear systems in high spatial dimensions. In particular, we discuss the role of multi-grid variants.

## 1 Introduction

In the standard setting, linear systems

$$Ax = b \tag{1.1}$$

are described by a regular matrix  $A \in \mathbb{R}^{n \times n}$  and vectors  $x, b \in \mathbb{R}^n$ . Often we are interested in large-scale problems with  $n$  about 100 000 to 100 000 000. In that case, vectors and matrices are represented differently. Vectors of this size can be described explicitly in the full format, i.e., each entry is stored. Matrices of the size  $n \times n$  must be represented by less than  $n^2$  units. In the fortunate case of sparse matrices, the sparse format is an easy remedy. For fully populated matrices one can use, e.g., the technique of hierarchical matrices resulting in a storage requirement of<sup>1</sup>  $\mathcal{O}(n \log^* n)$  (cf. [14, 17]).

The situation changes if  $n$  becomes much larger. A concrete example is the Lyapunov matrix equation. Consider a parabolic problem  $y_t = Ly + Bu$  with the elliptic operator  $L$  discretised by a matrix  $A \in \mathbb{R}^{n \times n}$  as considered above. The right-hand side contains the control variable  $u$ . Linear-quadratic optimal control problems leads to a description of the control involving the solution  $X \in \mathbb{R}^{n \times n}$  of a quadratic matrix Riccati equation (cf. Grasedyck–Hackbusch–Khoromskij [11]). For simplicity we consider its linearisation, the linear Lyapunov equation

$$AX + XA = B \quad (B \in \mathbb{R}^{n \times n} \text{ given, } X \in \mathbb{R}^{n \times n} \text{ sought}) \tag{1.2}$$

(cf. Benner–Breiten [3]). Even if  $A$  and  $B$  are sparse, the solution  $X$  is a fully populated matrix. Equation (1.2) is an unusual notation for a linear system with  $n^2$  equations involving a matrix  $\mathbf{A} \in \mathbb{R}^{n^2 \times n^2}$ :

$$\mathbf{A}\mathbf{x} = \mathbf{b}, \quad \mathbf{x}, \mathbf{b} \in \mathbb{R}^{n^2}, \quad \mathbf{A} \in \mathbb{R}^{n^2 \times n^2}. \tag{1.3}$$

Now the full representation of the vector  $\mathbf{x} \in \mathbb{R}^{n^2}$  is as problematic as the full representation of the fully populated matrix  $X \in \mathbb{R}^{n \times n}$ . This shows that for much larger sizes  $N := n^2$  the treatment of a linear system (1.1) must be changed substantially.

Concerning the spatial dimension, we add the following remark. Let  $L$  be a differential operator of a 3D problem, while  $A$  is the discrete version of  $L$ . Then the matrix  $\mathbf{A}$  in (1.3) discretises a six-dimensional pde. The matrix  $\mathbf{A}$  can be defined by means of the tensor product (here also called Kronecker product):

$$\mathbf{A} = I \otimes A + A \otimes I.$$

## 2 Tensors

The tensor space  $\mathbf{V} := \bigotimes_{j=1}^d V_j$  exists for all vector spaces  $V_j$  (details in Hackbusch [15, 16]). In the following, we discuss the tensor space for particular vector spaces  $V_j$ .

---

<sup>1</sup>The asterisk in  $\log^* n$  replaces a variable exponent. This factor also contains the bound of the local rank of the hierarchical method, which usually depends indirectly on  $n$ .

## 2.1 Grid functions ( $V_j = \mathbb{R}^{n_j}$ )

Consider a regular grid in  $d$  spatial dimensions. More precisely, let  $\omega_j = \{x_1^{(j)}, \dots, x_{n_j}^{(j)}\}$  be a regular grid consisting of  $n_j$  nodal points and define the *product grid*  $\omega = \omega_1 \times \omega_2 \times \dots \times \omega_d$ . The nodal points  $\mathbf{x} \in \omega$  are indexed by a  $d$ -tuple  $(i_1, \dots, i_d)$  with  $1 \leq i_j \leq n_j$ :  $\mathbf{x}[i_1, \dots, i_d] = (x_{i_1}^{(1)}, \dots, x_{i_d}^{(d)})$ . Grid functions  $u$  defined on  $\omega$  are characterised by the nodal values  $u[i_1, \dots, i_d]$ , i.e.,  $u$  belongs to  $\mathbb{R}^N$  with  $N := \#\omega = \prod_{j=1}^d n_j$ . The  $d$ -dimensional structure is described by the tensor product  $\mathbb{R}^{n_1} \otimes \mathbb{R}^{n_2} \otimes \dots \otimes \mathbb{R}^{n_d}$  which is isomorphic to  $\mathbb{R}^N$ :

$$\mathbb{R}^N \simeq \mathbb{R}^{n_1} \otimes \mathbb{R}^{n_2} \otimes \dots \otimes \mathbb{R}^{n_d}.$$

The tensor product of vectors  $v^{(j)} \in \mathbb{R}^{n_j}$  is defined by

$$\left( \bigotimes_{j=1}^d v^{(j)} \right) [i_1, i_2, \dots, i_d] := \prod_{j=1}^d v^{(j)}[i_j].$$

Usually, the dimension  $N := n^d$  (for  $n_j = n$ ) exceeds the available storage by far.

## 2.2 Functions ( $V_j = L^2(\Omega_j)$ )

Also functions  $f$  defined on a Cartesian product  $\Omega = \Omega_1 \times \Omega_2 \times \dots \times \Omega_d$  are tensors. For instance,  $L^2(\Omega)$  can be considered as the tensor space

$$L^2(\Omega) = L^2(\Omega_1) \otimes L^2(\Omega_2) \otimes \dots \otimes L^2(\Omega_d).$$

The tensor product of univariate functions is defined by

$$\left( \bigotimes_{j=1}^d f_j \right) (x_1, x_2, \dots, x_d) := \prod_{j=1}^d f_j(x_j). \quad (2.1)$$

Differently from the finite-dimensional case, the (topological) tensor space depends on the chosen norm.

## 2.3 Matrices ( $V_j = \mathbb{R}^{n_j \times m_j}$ )

A tensor space can also be generated by matrices, since the set of matrices in  $\mathbb{R}^{n_j \times m_j}$  is a particular vector space. Let  $A_j : V_j \rightarrow W_j$  ( $1 \leq j \leq d$ ) be linear mappings and form the tensor spaces  $\mathbf{V} = V_1 \otimes V_2 \otimes \dots \otimes V_d$  and  $\mathbf{W} = W_1 \otimes W_2 \otimes \dots \otimes W_d$ . The tensor product (*Kronecker product*)

$$\mathbf{A} = A_1 \otimes A_2 \otimes \dots \otimes A_d$$

can be interpreted as the linear map  $\mathbf{A} : \mathbf{V} \rightarrow \mathbf{W}$  defined by

$$\mathbf{A} : v^{(1)} \otimes v^{(2)} \otimes \dots \otimes v^{(d)} \mapsto A_1 v^{(1)} \otimes A_2 v^{(2)} \otimes \dots \otimes A_d v^{(d)} \quad (2.2)$$

for all  $v^{(j)} \in V_j$ . Using matrices  $A_j \in \mathbb{R}^{n \times n}$  of moderate size, one can construct huge matrices  $\mathbf{A} \in \mathbb{R}^{n^d \times n^d}$ .

## 2.4 $r$ -Term Format (Canonical Format)

By definition, any algebraic tensor  $\mathbf{v} \in \mathbf{V} = V_1 \otimes V_2 \otimes \dots \otimes V_d$  has a representation

$$\mathbf{v} = \sum_{\nu=1}^r v_\nu^{(1)} \otimes v_\nu^{(2)} \otimes \dots \otimes v_\nu^{(d)} \quad (2.3)$$

with  $v_\nu^{(j)} \in V_j$  and suitable  $r \in \mathbb{N}$ . Fixing some  $r \in \mathbb{N}$ , we set

$$\mathcal{R}_r := \left\{ \sum_{\nu=1}^r v_\nu^{(1)} \otimes v_\nu^{(2)} \otimes \dots \otimes v_\nu^{(d)} : v_\nu^{(j)} \in V_j \right\}.$$

The number  $r$  is called the *representation rank*<sup>2</sup> of  $\mathbf{v} \in \mathcal{R}_r$ . Obviously, the storage size for any  $\mathbf{v} \in \mathcal{R}_r$  is bounded by  $rdn$ , where  $n = \max_j \dim V_j$ . If  $r$  is of moderate size, this format is advantageous. Often, a tensor  $\mathbf{v}$  requires a high rank, but there is some approximation  $\mathbf{v}_\varepsilon \in \mathcal{R}_r$  with moderate  $r = r(\varepsilon)$ . An example will follow in §3.1.

Note that (2.3) holds for all vector spaces  $V_j$ . In the case of matrix spaces, the  $r$ -term representation of a matrix reads as

$$\mathbf{A} = \sum_{\nu=1}^r \bigotimes_{j=1}^d A_\nu^{(j)}, \quad A_\nu^{(j)} \in \mathbb{R}^{n_j \times m_j}. \quad (2.4)$$

## 2.5 Numerical Tensor Calculus

The numerical treatment of tensors is called ‘numerical tensor calculus’. Besides the representation and storage of tensors, the efficient performance of the tensor operations is required. Since the tensor space  $\mathbf{V}$  is a vector space, the addition is fundamental:

$$\mathbf{v} + \mathbf{w}, \quad \mathbf{A} + \mathbf{B}. \quad (2.5a)$$

Using operands of the form (2.3) or (2.4), the result has the same structure with increased representation rank. Note that this implementation of the addition requires no computational work.

Let  $\langle \cdot, \cdot \rangle_j$  be the scalar product in  $V_j$ . Then the *canonical scalar product*  $\langle \cdot, \cdot \rangle$  in  $\mathbf{V} = \bigotimes_{j=1}^d V_j$  is defined via

$$\left\langle \left( \bigotimes_{j=1}^d v^{(j)} \right), \left( \bigotimes_{j=1}^d w^{(j)} \right) \right\rangle = \prod_{j=1}^d \langle v^{(j)}, w^{(j)} \rangle_j. \quad (2.5b)$$

For instance, the Euclidean scalar products  $\langle \cdot, \cdot \rangle_j$  lead to the Euclidean scalar product  $\langle \mathbf{v}, \mathbf{w} \rangle = \sum_{\mathbf{i}} \mathbf{v}[\mathbf{i}] \mathbf{w}[\mathbf{i}]$  in  $\mathbf{V}$ . Using the tensor constructions for matrices and vectors, the *matrix-vector multiplication* is defined by

$$\left( \bigotimes_{j=1}^d A^{(j)} \right) \left( \bigotimes_{j=1}^d v^{(j)} \right) = \bigotimes_{j=1}^d A^{(j)} v^{(j)} \quad (2.5c)$$

(cf. (2.2)). The formulae (2.5b,c) show that the tensor operation applied to elementary tensors can easily be computed by operations related to the simpler vector spaces  $V_j$ . The same statement holds for further operations as the *Hadamard product* (entrywise product) and the *convolution*  $\mathbf{u} = \mathbf{v} \star \mathbf{w}$  with  $\mathbf{u}_{\mathbf{i}} = \sum_{\mathbf{0} \leq \mathbf{k} \leq \mathbf{i}} \mathbf{v}_{\mathbf{i}-\mathbf{k}} \mathbf{w}_{\mathbf{k}}$ .

## 2.6 Truncation

The representation rank<sup>3</sup>  $r$  is the critical quantity in the storage size  $rdn$ . Unfortunately, all operations have the property that their result has a (much) larger representation rank than their operands. As an example we consider the matrix-vector multiplication, where the matrix  $\mathbf{A}$  has the representation rank  $r_A$ , while the vector  $\mathbf{v}$  uses the representation rank  $r_v$ :

$$\mathbf{w} := \mathbf{A}\mathbf{v} = \left( \sum_{\nu=1}^{r_A} \bigotimes_{j=1}^d A_\nu^{(j)} \right) \left( \sum_{\mu=1}^{r_v} \bigotimes_{j=1}^d v_\mu^{(j)} \right) = \sum_{\nu=1}^{r_A} \sum_{\mu=1}^{r_v} \bigotimes_{j=1}^d A_\nu^{(j)} v_\mu^{(j)}.$$

Obviously, the product  $\mathbf{w}$  has enlarged the representation rank to  $r := r_A r_v$ . Repeating the operations, the calculations become more and more costly. A remedy is the *rank truncation*: the result  $\mathbf{w}$  has to be approximated by  $\mathbf{w}'$  with a much smaller rank  $r'$ . This truncation should be controlled, i.e., we should be able to estimate the truncation error.

<sup>2</sup>The representation rank has to be distinguished from  $\text{rank}(\mathbf{v})$ , the rank of the tensor  $\mathbf{v}$ , which is the minimal  $r$  such that  $\mathbf{v} \in \mathcal{R}_r$ . Since the determination of  $\text{rank}(\mathbf{v})$  is in general NP hard (cf. Håstad [20]), the representation rank used in practice will be larger than the  $\text{rank}(\mathbf{v})$ .

<sup>3</sup>The upper theoretical bound  $r \leq O(n^{d-1})$  is useless, since then  $rdn = O(n^d)$  is not better than the full representation by all entries.

Unfortunately, truncation within the format  $\mathcal{R}_r$  is a difficult procedure. Moreover, instabilities may occur. The reason is that  $\mathcal{R}_r$  is not closed and any approximation of  $\mathbf{v} \in \overline{\mathcal{R}_r} \setminus \mathcal{R}_r$  by a sequence  $\mathcal{R}_r \ni \mathbf{v}_n \rightarrow \mathbf{v}$  leads to numerical cancellation problems (cf. [15, §9.4.3]). Because of these shortcomings one is looking for other tensor representations.

In spite of the negative statement from above, we have to emphasise that the format  $\mathcal{R}_r$  is perfect if, by some method, we are able to find a suitable  $r$ -term representation. Here ‘suitable’ means stable and with moderate representation rank  $r$ . In (3.1) we shall see such an example.

## 2.7 Other Tensor Representations

An alternative traditional representation is the subspace representation or Tucker format (cf. Tucker [30]). Instead of a scalar representation rank we have a  $d$ -tuple  $\mathbf{r} = (r_1, \dots, r_d)$  of integers. In each vector space  $V_j$  we choose  $r_j$  vectors (usually, linearly independent ones):

$$b_1^{(j)}, b_2^{(j)}, \dots, b_{r_j}^{(j)} \in V_j \quad (2.6)$$

with the intention that  $r_j \ll n_j := \dim(V_j)$ . Then we try to represent or approximate a tensor by

$$\mathbf{v} = \sum_{1 \leq \ell_1 \leq r_1} \dots \sum_{1 \leq \ell_d \leq r_d} \mathbf{a}[\ell_1, \dots, \ell_d] b_{\ell_1}^{(1)} \otimes \dots \otimes b_{\ell_d}^{(d)} \quad (2.7)$$

This approach was originally introduced for the case of  $d = 3$ . For large  $d$ , this approach is rather costly since the involved *core tensor*  $\mathbf{a} \in \mathbb{R}^{r_1} \otimes \mathbb{R}^{r_2} \otimes \dots \otimes \mathbb{R}^{r_d}$  requires the storage size  $\prod_{j=1}^d r_j$ . On the other hand, this format has very favourable properties. The basis (2.6) can be chosen orthonormal and, using the so-called matricisation, the core tensor  $\mathbf{a}$  can be isomorphically mapped into a matrix. This allows us to use matrix techniques like the singular value decomposition. The arising ‘higher order SVD’ (HOSVD, cf. [6], [15, §8.3]) is a perfect tool for the truncation as required in §2.6.

A more recent format is the hierarchical format (cf. Hackbusch–Kühn [19] and [15, §11]). It provides an easy truncation, while the storage size  $dr^3 + dnr$  is linear in  $d$  ( $r$  is the maximum of a rank tuple). As long as  $r$  is bounded, a tensor size of  $n^d$  for  $n = d = 1000$  is harmless, since only the product  $dn$  appears. For later purpose, we mention that the typical tensor operations cost amounts to

$$\mathcal{O}(dr^4 + dnr^2) \quad (2.8)$$

elementary floating-point operations ( $r$  defined as above).

## 3 Linear Systems

### 3.1 Model Problem: Laplace Operator

We consider the  $d$ -dimensional Laplace operator in a cube,

$$-\Delta = - \sum_{j=1}^d \frac{\partial^2}{\partial x_j^2} \quad \text{in } \Omega = [0, 1]^d \subset \mathbb{R}^d,$$

with Dirichlet condition  $u = 0$  on  $\partial\Omega$ , and discretise by a finite difference scheme with the grid size  $h = 1/(n-1)$  in a regular product grid. Let  $A_j$  be the (positive definite) tridiagonal matrix built by the (one-dimensional) negative second divided differences. Then the system matrix has the form

$$\mathbf{A} = A_1 \otimes I \otimes I \otimes \dots \otimes I + I \otimes A_2 \otimes I \otimes \dots \otimes I + I \otimes I \otimes A_3 \otimes \dots \otimes I + \dots,$$

i.e., it is a  $d$ -term representation of the form (2.4). A similar result can be obtained for finite elements using a regular grid, but then the identity  $I$  is replaced with the mass matrix. The following considerations hold also for other positive definite matrices  $A_j$ .

One may be afraid that the huge system  $\mathbf{A}\mathbf{x} = \mathbf{b}$  has a huge condition number. This is not the case; instead, the condition is determined by the one-dimensional problems:

$$\min_{1 \leq j \leq d} \text{cond}(A_j) \leq \text{cond}(\mathbf{A}) \leq \max_{1 \leq j \leq d} \text{cond}(A_j).$$

Obviously, the spectrum of  $\mathbf{A}$  is contained in  $[a, \infty)$  for  $a := \sum_{j=1}^d \lambda_1^{(j)}$ , where  $\lambda_1^{(j)}$  is the smallest eigenvalue of  $A_j$ . Using a suitable scaling, we may assume that  $a = 1$ .

**Lemma 3.1** *There are (computable) coefficients  $\alpha_\nu, \beta_\nu > 0$  such that*

$$\mathbf{A}^{-1} \approx \mathbf{B}_r := \sum_{\nu=1}^r \alpha_\nu \bigotimes_{j=1}^d \exp(-\beta_\nu A_j) \in \mathcal{R}_r$$

with

$$\|\mathbf{A}^{-1} - \mathbf{B}_r\|_2 \leq C_1 \exp(-C_2 r^{1/2}).$$

Even  $\|\mathbf{A}^{-1} - \mathbf{B}_r\|_2 \leq C_1 \exp(-C_2 r)$  holds, if  $\sigma(\mathbf{A}) \subset [1, R]$ ,  $R < \infty$  ( $C_1, C_2$  depend on  $R$ ).

The proof uses the approximation of  $\frac{1}{x}$  in  $[a, \infty)$  by the exponential sum  $E_r(x) = \sum_{\nu=1}^r \alpha_\nu \exp(-\beta_\nu x)$ . Choosing the optimal coefficients  $\alpha_\nu, \beta_\nu$ , one proves the following error with respect to the maximum norm:

$$\left\| \frac{1}{x} - E_r(x) \right\|_{\infty, [1, \infty)} \leq \mathcal{O}(\exp(-cr^{1/2}))$$

(cf. Braess–Hackbusch [4, 5]). For positive definite matrices  $\mathbf{A}$  with  $\sigma(\mathbf{A}) \subset [1, \infty)$ ,  $E_r(\mathbf{A})$  approximates  $\mathbf{A}^{-1}$  with the same bound:  $\|E_r(\mathbf{A}) - \mathbf{A}^{-1}\|_2 \leq \left\| \frac{1}{x} - E_r(x) \right\|_{\infty, [1, \infty)}$ . In the case of  $\mathbf{A} = A_1 \otimes I \otimes \dots \otimes I + \dots + I \otimes \dots \otimes I \otimes A_d$  one has to evaluate  $\exp(-\beta_\nu \mathbf{A})$  and obtains

$$\mathbf{B}_r = E_r(\mathbf{A}) = \sum_{\nu=1}^r \alpha_\nu \bigotimes_{j=1}^d \exp(-\beta_\nu A_j) \in \mathcal{R}_r. \quad (3.1)$$

### 3.2 Solution of Linear Systems

Consider a linear system

$$\mathbf{A}\mathbf{x} = \mathbf{b},$$

where  $\mathbf{x}, \mathbf{b} \in \mathbf{V} = \bigotimes_{j=1}^d V_j$  and  $\mathbf{A} \in \bigotimes_{j=1}^d \mathcal{L}(V_j, V_j) \subset \mathcal{L}(\mathbf{V}, \mathbf{V})$  are represented in one of the formats. The general form of a linear iteration is

$$\mathbf{x}^{m+1} = \mathbf{x}^m - \mathbf{B}(\mathbf{A}\mathbf{x}^m - \mathbf{b})$$

with any matrix  $\mathbf{B}$  (cf. [12, (3.2.4)]). If this algorithm is applied, e.g., with the starting value  $\mathbf{x}^0 = 0$ , the representation ranks of  $\mathbf{x}^m$  would blow up. Therefore, according to §2.6, a truncation  $T$  must be applied. This yields the ‘truncated iteration’

$$\mathbf{x}^{m+1} = T(\mathbf{x}^m - \mathbf{B}(T(\mathbf{A}\mathbf{x}^m - \mathbf{b}))).$$

The cost per step is  $nd$  times powers of the involved representation ranks.

**Remark 3.2** *Let  $\mathbf{A}_\Delta$  be the discretisation of  $\Delta$  from above and  $\mathbf{B}_r \in \mathcal{R}_r$  the approximation of  $\mathbf{A}_\Delta^{-1}$  by (3.1). If the matrix  $\mathbf{A}$  corresponds to an elliptic pde of order 2,  $\mathbf{A}$  and  $\mathbf{A}_\Delta$  are spectrally equivalent (cf. [12, Lemma 8.4.1]). Therefore  $\mathbf{B} := \mathbf{B}_r \in \mathcal{R}_r$  is an appropriate preconditioner.*

The approach from above is, e.g., considered by Khoromskij [22].

### 3.3 cg-Like Methods

The exact cg recursion yields a sequence of conjugate (i.e.,  $\mathbf{A}$ -orthogonal) vectors. It is well known that because of the floating point errors, the orthogonality is lost after several iterations. This effect is even worse for the conjugate gradient algorithm combined with tensor rank truncations, since the relative truncation errors are usually larger than the machine precision. There are two remedies:

1. The cg method is restricted to few iterations so that the loss of orthogonality is still weak enough.
2. As soon as necessary, a re-orthogonalisation is performed.

The first approach requires fast convergence, i.e., a very good preconditioning. The second approach has a drawback which will be discussed next. First we mention the following articles about conjugate gradient methods in the context of tensor methods: Tobler [29], Kressner–Tobler [23, 24, 25], and Savas–Eldén [28].

Assume that  $\mathbf{u}_k$  ( $1 \leq k \leq m$ ) are tensors belonging to  $\mathcal{R}_r$  for a fixed  $r$ . The orthonormalisation changes  $\mathbf{u}_2$  into a linear combination  $\tilde{\mathbf{u}}_2 = \alpha \mathbf{u}_1 + \beta \mathbf{u}_2$ . Obviously, the result  $\tilde{\mathbf{u}}_2$  belongs to  $\mathcal{R}_{2r}$ . In general, the orthonormalisation process produces  $\tilde{\mathbf{u}}_k \in \mathcal{R}_{kr}$  with increasing representation rank  $kr$ . A reduction of the rank by a truncation is not possible without losing again orthonormality.

To avoid this increase of the rank, Ballani–Grasedyck [2] propose a variant of the GMRES method with tensor truncation and without orthonormalisation. Nevertheless, the iterates are built in such a way that the residuals are decreasing.

## 4 Multigrid Approach

First we transfer the setting of the geometric multigrid method to the tensor case. In §4.7 we discuss the pros and cons of the method.

### 4.1 Grid Hierarchy

Since the tensors directly correspond to grid functions associated to product grids (cf. §2.1), the traditional geometric multigrid structure is appropriate.

Let  $V_j^0, V_j^1, \dots, V_j^\ell, \dots$  be a one-dimensional grid hierarchy. In the case of a difference scheme,  $V_j^\ell = \mathbb{R}^{n_\ell^{(j)}}$  corresponds to a grid with  $n_\ell^{(j)}$  points, where  $n_0^{(j)} < n_1^{(j)} < \dots$ . In the case of a finite element discretisation,  $V_j^\ell$  may be the space of piecewise linear functions defined on a grid of  $n_\ell^{(j)}$  points.

The  $d$ -dimensional grid hierarchy is given by the tensor spaces

$$\mathbf{V}^0, \mathbf{V}^1, \dots, \mathbf{V}^\ell, \dots \quad \text{with} \quad \mathbf{V}^\ell := \bigotimes_{j=1}^d V_j^\ell.$$

In the case of the difference scheme, the elements of  $\mathbf{V}^\ell$  are grid functions on a  $d$ -dimensional grid with  $N := \prod_{j=1}^d n_\ell^{(j)}$  nodal points (cf. §2.1), while, in the finite element case,  $\mathbf{V}^\ell$  consists of piecewise  $d$ -linear functions on the cubical grid of the dimension  $N := \prod_{j=1}^d n_\ell^{(j)}$  (cf. §2.2). Note that the corresponding finite element basis functions are elementary tensors of the form (2.1).

### 4.2 Prolongation, Restriction

Let  $p_j : V_j^\ell \rightarrow V_j^{\ell+1}$  be the one-dimensional linear prolongation (cf. [13, §3.4.2]). The  $d$ -linear prolongation  $\mathbf{p} : \mathbf{V}^\ell \rightarrow \mathbf{V}^{\ell+1}$  is the elementary (Kronecker) tensor product of the one-dimensional linear prolongations:

$$\mathbf{p} = p_1 \otimes p_2 \otimes \dots \otimes p_d.$$

Similarly, the restriction satisfies

$$\mathbf{r} = r_1 \otimes r_2 \otimes \dots \otimes r_d.$$

Note that usually  $\mathbf{r}$  is the adjoint of  $\mathbf{p}$  (cf. [13, §3.5]).

An important observation is the fact that the application of  $\mathbf{p}$  and  $\mathbf{r}$  does not increase the representation ranks. Let  $\delta \mathbf{x}^{\ell-1}$  be the correction obtained from the coarse grid. Then the multigrid iteration determines the new iterate by  $\mathbf{x}^\ell \leftarrow \mathbf{x}^\ell - \mathbf{p} \delta \mathbf{x}^{\ell-1}$ . To restrict the representation rank, we have to replace this assignment by the truncated version

$$\mathbf{x}^\ell \leftarrow T(\mathbf{x}^\ell - \mathbf{p} \delta \mathbf{x}^{\ell-1}).$$

### 4.3 Coarse-Grid Matrices

Let  $\mathbf{A}^L$  be the fine-grid matrix ( $L$  is the level number of the finest grid). The auxiliary matrices  $\mathbf{A}^\ell$  ( $\ell < L$ ) can be defined by the Galerkin approach (cf. [13, §3.7]):

$$\mathbf{A}^{\ell-1} := \mathbf{r} \mathbf{A}^\ell \mathbf{p} \quad \text{for } \ell = L, L-1, \dots, 1.$$

Assume that the fine-grid matrix  $\mathbf{A}^L$  has the representation

$$\mathbf{A}^L = \sum_{\nu=1}^{r_A} \bigotimes_{j=1}^d A_{\nu}^{L,j}.$$

The Galerkin approach leads to

$$\mathbf{A}^\ell = \sum_{\nu=1}^{r_A} \bigotimes_{j=1}^d A_{\nu}^{\ell,j}$$

with the same representation rank  $r_A$  involving

$$A_{\nu}^{L,j} := r_j A_{\nu}^{\ell,j} p_j.$$

The defect  $\mathbf{d}^{(\ell-1)} := \mathbf{A}^{\ell-1} \mathbf{x}^{\ell-1} - \mathbf{b}^{\ell-1}$  computed in the coarse grid has an increased rank because of the matrix-vector multiplication (see the example in §2.6). Hence, this step has to be replaced by the truncated version

$$\mathbf{d}^{(\ell-1)} := T(\mathbf{A}^{\ell-1} \mathbf{x}^{\ell-1} - \mathbf{b}^{\ell-1}).$$

### 4.4 Smoothing

#### 4.4.1 Richardson Iteration

The exact Richardson iteration is

$$\mathbf{x}_{\text{new}}^\ell := \mathbf{x}_{\text{old}}^\ell - \omega_\ell (\mathbf{A}^\ell \mathbf{x}_{\text{old}}^\ell - \mathbf{b}^\ell).$$

As discussed in §3.2, this approach is not advisable in the tensor case because of the increasing representation ranks. Instead, the *truncated* Richardson iteration

$$\mathbf{x}_{\text{new}}^\ell := T(\mathbf{x}_{\text{old}}^\ell - \omega_\ell (\mathbf{A}^\ell \mathbf{x}_{\text{old}}^\ell - \mathbf{b}^\ell))$$

is the method of choice.

A possible improvement is the semilinear variant of the truncated Richardson iteration (cf. [13, §3.3.5]).

#### 4.4.2 Jacobi Iteration

The damped Jacobi iteration is an often applied smoothing iteration. However, in the tensor case, the performance of this iteration is already too complicated as pointed out below.

The diagonal of the (Kronecker) matrix  $\mathbf{A}^\ell = \sum_{\nu=1}^{r_A} \bigotimes_{j=1}^d A_{\nu}^{\ell,j}$  is given by

$$\mathbf{D}^\ell := \text{diag}\{\mathbf{A}^\ell\} = \sum_{\nu=1}^{r_A} \bigotimes_{j=1}^d D_{\nu,j}^{\ell} \quad \text{with } D_{\nu,j}^{\ell} := \text{diag}\{A_{\nu,j}^{\ell}\}.$$

In the tensor case, diagonal matrices may be called sparse because of the many zero entries, but in fact they are not sparse since the number of nonzero elements is still huge. The number of nonzero entries of  $\mathbf{D}^\ell \in \mathbb{R}^{N \times N}$  is  $N = \prod_{j=1}^d n_\ell^{(j)}$ , which is far too large for practical use. Instead one may try to use an approximation

$$(\mathbf{D}^\ell)^{-1} \approx \mathbf{B}_r := E_r(\mathbf{D}^\ell) = \sum_{\mu=1}^r \alpha_\mu \exp(-\beta_\mu \mathbf{D}^\ell)$$

as in (3.1). However, Kronecker exponentials of the form  $\exp(\bigotimes_{j=1}^d D_j)$  are not easy to evaluate or to apply to a tensor. The only exception arises when at most one  $D_j$  is different from the identity  $I$ .

## 4.5 Treatment of the Coarsest Grid

We recall that  $\dim(\mathbf{V}^\ell) = \prod_{j=1}^d n_j^{(\ell)}$  with  $n_j^{(\ell)} := \dim(V_j^\ell)$ . Since the dimension of the coarsest grid is  $\prod_{j=1}^d n_j^{(0)}$ , a *direct* solution of the system at the lowest level requires

- either  $n_j^{(0)} = 1$ ,
- or  $n_j^{(0)}$  and  $d$  so small that also  $\prod_{j=1}^d n_j^{(0)}$  is sufficiently small.

The choice  $n_j^{(0)} = 1$  is possible for positive definite problems (cf. [13, §7.2]). Only if  $n_j^{(0)} = 1$ , the coarse-grid problem has a size  $\prod_{j=1}^d n_j^{(0)} = 1$  not depending on  $d$ . Otherwise, some other method from §§3.2–3.3 must be applied.

## 4.6 Convergence

Choosing the multigrid parameters as above, we can apply the standard convergence results (e.g., [13, Theorems 7.2.2 and 7.2.3]), provided that no tensor truncation is applied. The additional truncation effect is similar to the truncated iteration analysed by Hackbusch–Khoromskij–Tyrtshnikov [18] (see also [14, §14.3.2] or [17, §15.3.2]).

Numerical examples for the use of a tensor multigrid iteration can be seen in Ballani–Grasedyck [2, Example 7.5]. These examples with  $n_{\text{fine}} := n^{(L)} = 1023$  and dimensions up to  $d = 32$  demonstrate that the convergence behaviour does not depend on  $d$ .

In (1.2) we mention the Lyapunov equation. A multigrid solution of the more general Sylvester matrix equation  $AX - XB = C$  is described in Grasedyck–Hackbusch [10]. A nonlinear multigrid approach to the quadratic Riccati equation is given in [9].

## 4.7 Cost

A disadvantage is the following effect. In traditional 3D multigrid applications, computations at level  $\ell - 1$  with the step size  $h_{\ell-1} = 2h_\ell$  cost only  $\frac{1}{8}$  of the computations at level  $\ell$ . This explains why the cost of one V- or W-cycle is dominated by the operations spent on the finest grid.

In tensor applications, the cost of operations is described in (2.8):  $\mathcal{O}(dr^4 + dnr^2)$ . In the multigrid case,  $n$  has to be replaced with  $n^{(\ell)} := \max\{n_j^{(\ell)} : 1 \leq j \leq d\}$ . For the finest grid, we use the notation  $n_{\text{fine}} := n^{(L)}$ . Assuming  $n_j^{(\ell)} \approx 2n_j^{(\ell-1)}$ , we obtain the following work per V- or W-cycle:

$$\begin{aligned} \text{V-cycle: } & \mathcal{O}(dr^4L + dn_{\text{fine}}r^2), \\ \text{W-cycle: } & \mathcal{O}(dr^42^L + dn_{\text{fine}}r^2L) = \mathcal{O}(dr^4n_{\text{fine}} + dn_{\text{fine}}r^2L). \end{aligned}$$

This result can be interpreted in different ways. On the negative side, the portion of the work corresponding to the auxiliary grids  $\ell < L$  is larger, but on the positive side, even for the W-cycle, the cost is less than  $\mathcal{O}(n_{\text{fine}}^{1+\varepsilon})$  for all  $\varepsilon > 0$  with respect to  $n_{\text{fine}}$ . Nevertheless, the factor  $r^4$  may become more problematic than  $n_{\text{fine}}$ .

A possible reduction of the cost may be obtained by choosing different representation ranks  $r$  for the finest grid and for the auxiliary problems at the levels  $\ell < L$ . For  $\ell = L$  a possibly larger rank  $r_{\text{fine}}$  is needed to approximate the solution of the discretised pde. The size of  $r_{\text{fine}}$  depends on the nature of the solution. However, for  $\ell < L$  only corrections are to be represented and the rank required by the corrections is not related to  $r_{\text{fine}}$ .

## 4.8 Parabolic Problems

Since the dimensions in the sense of number of coordinates are no limitation, space-time simultaneous discretisations with the additional time variable are not disadvantageous. In this respect, the results of Andreev–Tobler [1] are of interest. In the latter paper, a BPX preconditioner is used.

## 5 Variational Approach

Finally we mention a quite different approach to solving  $\mathbf{Ax} = \mathbf{b}$  approximately. If  $\mathbf{A}$  is positive definite, we may minimise the quadratic cost function

$$\Phi(\mathbf{x}) := \langle \mathbf{Ax}, \mathbf{x} \rangle - 2 \langle \mathbf{b}, \mathbf{x} \rangle .$$

In the general case of a regular  $\mathbf{A}$ , define

$$\Phi(\mathbf{x}) := \|\mathbf{Ax} - \mathbf{b}\|^2 \quad \text{or} \quad \Phi(\mathbf{x}) := \|\mathbf{B}(\mathbf{Ax} - \mathbf{b})\|^2$$

with a suitable preconditioner  $\mathbf{B}$  and try to minimise  $\Phi(\mathbf{x})$ . The minimisation over all tensors  $\mathbf{x}$  is not feasible because of the huge dimension. Instead one fixes a certain format for the representation of  $\mathbf{x}$  and minimises over all representation parameters of  $\mathbf{x}$ . In practice, this works quite well although the theoretical understanding of the convergence properties is still incomplete. Another difficulty is that  $\Phi$  has many local minima and the minimisation is nonconvex.

Concerning the variational approach we refer to the following papers: Espig–Hackbusch–Rohwedder–Schneider [7], Falcó–Nouy [8], Holtz–Rohwedder–Schneider [21], Mohlenkamp [26], Osedelets [27] and others cited in these papers.

## References

- [1] Andreev, R., Tobler, C.: Multilevel preconditioning and low-rank tensor iteration for space-time simultaneous discretizations of parabolic PDEs. *Numer. Linear Algebra Appl.* (2014). Published on-line
- [2] Ballani, J., Grasedyck, L.: A projection method to solve linear systems in tensor format. *Numer. Linear Algebra Appl.* **20**, 27–43 (2013)
- [3] Benner, P., Breiten, T.: Low rank methods for a class of generalized Lyapunov equations and related issues. *Numer. Math.* **124**, 441–470 (2013)
- [4] Braess, D., Hackbusch, W.: Approximation of  $1/x$  by exponential sums in  $[1, \infty)$ . *IMA J. Numer. Anal.* **25**, 685–697 (2005)
- [5] Braess, D., Hackbusch, W.: On the efficient computation of high-dimensional integrals and the approximation by exponential sums. In: R.A. DeVore, A. Kunoth (eds.) *Multiscale, Nonlinear and Adaptive Approximation*, pp. 39–74. Springer, Berlin (2009)
- [6] De Lathauwer, L., De Moor, B., Vandewalle, J.: A multilinear singular value decomposition. *SIAM J. Matrix Anal. Appl.* **21**, 1253–1278 (2000)
- [7] Espig, M., Hackbusch, W., Rohwedder, T., Schneider, R.: Variational calculus with sums of elementary tensors of fixed rank. *Numer. Math.* **122**, 469–488 (2012)
- [8] Falcó, A., Nouy, A.: Proper generalized decomposition for nonlinear convex problems in tensor Banach spaces. *Numer. Math.* **121**, 503–530 (2012)
- [9] Grasedyck, L.: Nonlinear multigrid for the solution of large-scale Riccati equations in low-rank and  $\mathcal{H}$ -matrix format, *Numer. Linear Algebra Appl.* **15**, 779–807 (2008)
- [10] Grasedyck, L., Hackbusch, W.: A multigrid method to solve large scale Sylvester equations, *SIAM J. Matrix Anal. Appl.* **29**, 870–894 (2007)
- [11] Grasedyck, L., Hackbusch, W., Khoromskij, B.: Solution of large scale algebraic matrix Riccati equations by use of hierarchical matrices. *Computing* **70**, 121–165 (2003)
- [12] Hackbusch, W.: *Iterative Solution of Large Sparse Systems of Equations*, Springer, New York (1994)
- [13] Hackbusch, W.: *Multi-grid Methods and Applications*, *SCM*, vol. 4. Springer, Berlin (2003)

- [14] Hackbusch, W.: Hierarchische Matrizen - Algorithmen und Analysis. Springer, Berlin (2009)
- [15] Hackbusch, W.: Tensor Spaces and Numerical Tensor Calculus, *SCM*, vol. 42. Springer, Berlin (2012)
- [16] Hackbusch, W.: Numerical tensor calculus. *Acta Numerica* **23**, 651–742 (2014)
- [17] Hackbusch, W.: Hierarchical Matrices - Algorithms and Analysis. Springer, Berlin (2015). To appear
- [18] Hackbusch, W., Khoromskij, B., Tyrtyshnikov, E.E.: Approximate iterations for structured matrices. *Numer. Math.* **109**, 365–383 (2008)
- [19] Hackbusch, W., Kühn, S.: A new scheme for the tensor representation. *J. Fourier Anal. Appl.* **15**, 706–722 (2009)
- [20] Håstad, J.: Tensor rank is NP-complete. *J. Algorithms* **11**, 644–654 (1990)
- [21] Holtz, S., Rohwedder, T., Schneider, R.: The alternating linear scheme for tensor optimization in the tensor train format. *SIAM J. Sci. Comput.* **34**, A683–A713 (2012)
- [22] Khoromskij, B.: Tensor-structured preconditioners and approximate inverse of elliptic operators in  $\mathbb{R}^d$ . *Constr. Approx.* **30**, 599–620 (2009)
- [23] Kressner, D., Tobler, C.: Krylov subspace methods for linear systems with tensor product structure. *SIAM J. Matrix Anal. Appl.* **31**, 1688–1714 (2010)
- [24] Kressner, D., Tobler, C.: Low-rank tensor Krylov subspace methods for parametrized linear systems. *SIAM J. Matrix Anal. Appl.* **32**, 1288–1316 (2011)
- [25] Kressner, D., Tobler, C.: Preconditioned low-rank methods for high-dimensional elliptic PDE eigenvalue problems. *Comput. Methods Appl. Math.* **11**, 363–381 (2011)
- [26] Mohlenkamp, M.J.: Musing on multilinear fitting. *Linear Algebra Appl.* **438**, 834–852 (2013)
- [27] Oseledets, I.V.: DMRG approach to fast linear algebra in the TT-format. *Comput. Methods Appl. Math.* **11**, 382–393 (2011)
- [28] Savas, B., Eldén, L.: Krylov-type methods for tensor computations I. *Linear Algebra Appl.* **438**, 891–918 (2013)
- [29] Tobler, C.: Low-rank tensor methods for linear systems and eigenvalue problems. Doctoral thesis, ETH Zürich (2012)
- [30] Tucker, L.R.: Some mathematical notes on three-mode factor analysis. *Psychometrika* **31**, 279–311 (1966)