Max-Planck-Institut
für Mathematik
in den Naturwissenschaften
Leipzig

Existence and Computation of a Low Kronecker-Rank Approximant to the Solution of a Tensor System with Tensor Right-Hand Side

by

Lars Grasedyck

Preprint no.: 48 2003

Existence and Computation of a Low Kronecker-Rank Approximant to the Solution of a Tensor System with Tensor Right-Hand Side

L. Grasedyck,
Max-Planck-Institute for Mathematics in the Sciences,
Inselstr. 22-26, 04301 Leipzig, Germany

May 28, 2003

Abstract

In this paper we construct an approximation to the solution x of a linear system of equations $Ax = b$ of tensor product structure as it typically arises for finite element and finite difference discretisations of partial differential operators on tensor grids. For a right-hand side b of tensor product structure we can prove that the solution x can be approximated by a sum of $\mathcal{O}(\log(\varepsilon)^2)$ tensor product vectors where ε is the relative approximation error. Numerical examples for systems of size 1024^{256} indicate that this method is suitable for high-dimensional problems.

Key words: Data-sparse approximation, Sylvester equation, low rank approximation, Kronecker product, high-dimensional problems

1 INTRODUCTION

A general linear system of equations

$$Ax = b, \qquad A \in \mathbb{R}^{N\times N}, b \in \mathbb{R}^N,$$

can be solved with algorithms of complexity $\mathcal{O}(N^3)$, e.g., by Householder. This complexity can be reduced if, e.g., blockwise Gaussian elimination is possible. In that case Strassen's algorithm [13] yields a better order $\mathcal{O}(N^{\log_2(7)})$. Obviously, $\mathcal{O}(N^2)$ is a lower bound if no structure on the matrix A is imposed. For certain sparse systems it is possible to approximate the solution x by iterative schemes of complexity $\mathcal{O}(N)$ per step. Obviously, $\mathcal{O}(N)$ is a lower bound if no structure on the right-hand side b is imposed.

In this paper we consider linear systems of equations where the matrix A and the right-hand side b are of a special structure. Let $N = n^d$ denote the number of columns and rows of A. The right-hand side is given in tensor structure

$$b = \bigotimes_{i=1}^{d} b_i, \qquad b_i \in \mathbb{R}^n, \quad b_j = \prod_{i=1}^{d} (b_i)_{j_i} \quad \text{for } j \in \{1,\dots,n\}^d. \tag{1}$$

The matrix A possesses the tensor structure

$$A = \sum_{i=1}^{d} \widehat{A}_i, \qquad \widehat{A}_i = \underbrace{I \otimes \cdots \otimes I}_{i-1 \text{ terms}} \otimes A_i \otimes \underbrace{I \otimes \cdots \otimes I}_{d-i \text{ terms}}, \quad A_i \in \mathbb{R}^{n\times n} \tag{2}$$

with spectrum $\sigma(A)$ contained in the left complex halfplane. In the last Section 8 we discuss where such a structure may arise. Our algorithm can solve an equation of the structure (2) for a right-hand side of the form (1) with complexity $\mathcal{O}(dn\log(n)^2\log(\varepsilon)^{7/2})$ such that the approximant fulfils

$$\|x-\tilde{x}\|_2 \leq \varepsilon\|x\|_2.$$

The rest of this paper is organised as follows:

Section 2 recapitulates the results in the case $d = 2$ (Sylvester equation). Different methods for the computation of an approximant to x are compared.

The main approximation result is derived in Section 4. We prove that the solution x can be approximated by a vector $\tilde{x}$ which is the sum of vectors in the tensor structure (1).

In Section 7 we address the problem of computing the matrix exponential $\exp(tA_i)$ which is needed in the representation formula for the approximant $\tilde{x}$.

The last Section 8 presents numerical results for problems of dimension $d = 256$ and $n = 1024$. Since the full solution vector x has 2^{2560} entries, we can neither compare our results to methods from the literature nor can we compute the approximation error $\|x-\tilde{x}\|_2$ exactly. Instead, we estimate the approximation error by evaluation in few random entries.

2 PREVIOUS WORKS

The only known previous works are those for the case $d = 2$. There, a two-dimensional tensor vector $b_1 \otimes b_2$ can be identified with the rank 1 matrix $B := b_1 b_2^T$. The system

$$(A_1 \otimes I \;+\; I \otimes A_2)\, x = b$$

can be rewritten as a (matrix) Sylvester equation

$$A_2^T X + X A_1 = B,$$

where the sought solution X and the right-hand side B are matrices. The tensor form (1) of the right-hand side b implies that B is of rank at most 1. In [5] a proof for the existence of a rank k approximant X_k to the solution X is given, where the rank k necessary to achieve an approximation error of

$$\|X - X_k\|_2 \leq \varepsilon\|X\|_2$$

is proportional to $\log(\varepsilon)$. Of course, the estimate depends on the spectrum of A_1 and A_2, but it is independent of B.

A low rank approximation X_k to X can be computed in different ways:

1. Iterative methods like the ADI or Smith iteration (see [11]) can be performed exactly for few iterative step such that the ith iterate is of rank at most i.

2. The sign function in combination with hierarchical matrices is used in [7] to efficiently compute a low rank or hierarchical matrix approximation to X.

3. In a forthcoming paper [4] we explain how multigrid methods in combination with truncated singular value decompositions can be used to compute a best approximation X_k of rank at most k to the solution X of the Sylvester equation.

For higher dimensions $d > 2$ the term "best approximation of rank at most k" means a best approximation consisting of k vectors in the tensor form (1). In two dimensions $d = 2$ the singular value decomposition is a useful tool to compute such a best approximation which is missing for the case $d > 2$.

3 INVERSE OF A TENSOR MATRIX

Lemma 1 *Let $M \in \mathbb{R}^{n\times n}$. If the spectrum of M is contained in the left complex halfplane, i.e.,*

$$\sigma(M) \quad \subset \quad \mathbb{C}_- := \{x + iy \in \mathbb{C} \mid x < 0\}, \tag{3}$$

then the inverse to M is

$$M^{-1} = -\int_0^\infty \exp(tM)\mathrm{d}t. \tag{4}$$

Proof: $M\left(-\int_0^\infty \exp(tM)\mathrm{d}t\right) = -\int_0^\infty \frac{\partial}{\partial t}\exp(tM)\mathrm{d}t = \exp(0M) = I.$ ■

Lemma 2 *Let A be a tensor matrix of the structure (2). If the sum of the spectra of the A_i (which is the spectrum of A) is contained in the left complex halfplane, then the inverse to A is*

$$A^{-1} = -\int_0^\infty \bigotimes_{i=1}^{d} \exp(tA_i)\mathrm{d}t. \tag{5}$$

Proof: Application of Lemma 1 yields (5) since for each $t > 0$

$$\exp(tA) = \exp(t\sum_{i=1}^{d}\widehat{A}_i) \overset{\widehat{A}_i \text{ commute}}{=} \prod_{i=1}^{d}\exp(t\widehat{A}_i) = \bigotimes_{i=1}^{d}\exp(tA_i).$$

■

In the previous Lemma we exploited the commutativity of the $\widehat{A}_i$ from (2). In the context of finite element discretisations the matrix A is often of the structure

$$A^{FEM} = \sum_{i=1}^{d}\widehat{A}_i^{FEM}, \qquad \widehat{A}_i^{FEM} = M_1 \otimes \cdots \otimes M_{i-1} \otimes A_i \otimes M_{i+1} \otimes \cdots \otimes M_d, \quad M_i, A_i \in \mathbb{R}^{n_i\times n_i}, \tag{6}$$

with the so-called mass matrices M_i, such that the matrices $\widehat{A}_i^{FEM}$ do not necessarily commute. In this case we can derive a representation formula similar to (5) for the inverse to A^{FEM}.

Lemma 3 *Let A^{FEM} be a tensor matrix of the structure (6) with regular M_i. If the sum of the spectra of the $M_i^{-1}A_i$ is contained in the left complex halfplane, then the inverse to A^{FEM} is*

$$(A^{FEM})^{-1} = -\int_0^\infty \bigotimes_{i=1}^{d} \exp(tM_i^{-1}A_i)M_i^{-1}\mathrm{d}t. \tag{7}$$

Proof: We can factorise the matrix A^{FEM} into

$$A^{FEM} = \bigotimes_{i=1}^{d} M_i \ \cdot \ \widetilde{A}^{FEM},$$

where the matrix $\widetilde{A}^{FEM}$ is

$$\widetilde{A}^{FEM} = \sum_{i=1}^{d} \widetilde{A}_i, \qquad \widetilde{A}_i = \underbrace{I \otimes \cdots \otimes I}_{i-1 \text{ terms}} \otimes M_i^{-1}A_i \otimes \underbrace{I \otimes \cdots \otimes I}_{d-i \text{ terms}}.$$

By assumption both factors are regular and the inverse in factorised form yields (7):

$$\begin{aligned}(A^{FEM})^{-1} &= (\bigotimes_{i=1}^{d} M_i \ \cdot \ \widetilde{A}^{FEM})^{-1} = (\widetilde{A}^{FEM})^{-1} \bigotimes_{i=1}^{d} M_i^{-1} \\ &\overset{L.2}{=} -\int_0^\infty \bigotimes_{i=1}^{d} \exp(tM_i^{-1}A_i)\mathrm{d}t \ \cdot \ \bigotimes_{i=1}^{d} M_i^{-1} = -\int_0^\infty \bigotimes_{i=1}^{d} \exp(tM_i^{-1}A_i)M_i^{-1}\mathrm{d}t.\end{aligned}$$

■

The representation formula (7) involves an improper integral. For the numerical computation of an approximate inverse we have to apply a suitable quadrature formula. In the next Section we shall even find an exponentially convergent one.

4 LOW RANK APPROXIMATION

For the discretisation of the integral (5) we use the quadrature formula of Stenger [12]. The proof of the following Lemma can be found in [7] or derived from [12, Example 4.2.11].

Lemma 4 (Stenger) *Let $z \in \mathbb{C}$ with $\Re e(z) \le -1$. Then for each $k \in \mathbb{N}$ the quadrature points and weights*

$$\begin{aligned} h_{\mathrm{st}} &:= \pi^2/\sqrt{k} \\ t_j &:= \log\left(\exp(jh_{\mathrm{st}}) + \sqrt{1+\exp(2jh_{\mathrm{st}})}\right), \qquad (8) \\ w_j &:= h_{\mathrm{st}}/\sqrt{1+\exp(-2jh_{\mathrm{st}})} \qquad (9)\end{aligned}$$

fulfil

$$\left| \int_0^\infty \exp(tz)\mathrm{d}t - \sum_{j=-k}^{k} w_j \exp(t_j z) \right| \le C_{\mathrm{st}} \exp(|\Im m(z)|/\pi) \exp(-\pi\sqrt{2k}), \tag{10}$$

with a constant C_{st} independent of z and k.

The result of the previous Lemma for the scalar case can be transferred to the matrix case in the following Lemma. There we make use of the Dunford-Cauchy representation of the matrix exponential: for all $t \in \mathbb{R}$ and all matrices M with spectrum contained in the interior of an index 1 path Γ there holds

$$\exp(tM) = \frac{1}{2\pi i} \oint_\Gamma \exp(t\lambda)(\lambda I - M)^{-1} \mathrm{d}_\Gamma \lambda. \tag{11}$$

Lemma 5 *Let M be a matrix with spectrum $\sigma(M)$ contained in the strip $\Omega := -[2, \Lambda] \oplus i[-\mu, \mu] \subseteq \mathbb{C}_-$. Let Γ denote the boundary of $-[1, \Lambda + 1] \oplus i[-\mu - 1, \mu + 1]$. Then the quadrature points and weights from (8) and (9) fulfil*

$$\left\| \int_0^\infty \exp(tM) \mathrm{d}t - \sum_{j=-k}^{k} w_j \exp(t_j M) \right\| \leq \frac{C_{\mathrm{st}}}{2\pi} \exp(\frac{\mu + 1}{\pi} - \pi\sqrt{2k}) \oint_\Gamma \left\| (\lambda I - M)^{-1} \right\| \mathrm{d}_\Gamma \lambda. \tag{12}$$

In the case that M is symmetric, this simplifies to

$$\left\| \int_0^\infty \exp(tM) \mathrm{d}t - \sum_{j=-k}^{k} w_j \exp(t_j M) \right\|_2 \leq \frac{C_{\mathrm{st}}}{2\pi} \exp(\frac{1}{\pi} - \pi\sqrt{2k})(4 + 2\Lambda). \tag{13}$$

Proof:

$$\begin{aligned}
& \left\| \int_0^\infty \exp(tM) \mathrm{d}t - \sum_{j=-k}^{k} w_j \exp(t_j M) \right\| \\
\overset{(11)}{=} \; & \frac{1}{2\pi} \left\| \int_0^\infty \oint_\Gamma \exp(t\lambda)(\lambda I - M)^{-1} \mathrm{d}_\Gamma \lambda \mathrm{d}t - \sum_{j=-k}^{k} w_j \oint_\Gamma \exp(t_j \lambda)(\lambda I - M)^{-1} \mathrm{d}_\Gamma \lambda \right\| \\
\overset{(10)}{\leq} \; & \frac{C_{\mathrm{st}}}{2\pi} \exp(\frac{\mu + 1}{\pi} - \pi\sqrt{2k}) \oint_\Gamma \|(\lambda I - M)^{-1}\| \mathrm{d}_\Gamma \lambda.
\end{aligned}$$

In the symmetric case the spectrum of M is contained in the interval $-[2, \Lambda]$ ($\mu = 0$). The length of Γ is $4 + 2\Lambda$. Since the distance of Γ to $\sigma(M)$ is at least 1, we conclude $\|(\lambda I - M)^{-1}\|_2 \leq 1$ which yields (13). ∎

Lemma 6 *Let A be a matrix of the tensor structure (2) with spectrum $\sigma(A)$ contained in the strip $\Omega := -[\lambda_{min}, \lambda_{max}] \oplus i[-\mu, \mu] \subseteq \mathbb{C}_-$. Let b be the tensor vector (1). Let $k \in \mathbb{N}$ and t_j, w_j denote the points and weights from Lemma 4. Then the solution x to $Ax = b$ can be approximated by*

$$\tilde{x} := -\sum_{j=-k}^{k} \frac{2w_j}{\lambda_{min}} \bigotimes_{i=1}^{d} \exp\left(\frac{2t_j}{\lambda_{min}} A_i \right) b_i \tag{14}$$

with approximation error

$$\|x - \tilde{x}\| \leq \frac{C_{\mathrm{st}}}{\pi \lambda_{min}} \exp(\frac{2\mu\lambda_{min}^{-1} + 1}{\pi} - \pi\sqrt{2k}) \oint_\Gamma \left\| (\lambda I - 2A/\lambda_{min})^{-1} \right\| \mathrm{d}_\Gamma \lambda \;\; \|b\|. \tag{15}$$

Let A^{FEM} be a matrix of the tensor structure (6) and let the sum of the spectra of the $M^{-1}A_i$ be contained in Ω. Then the solution x to $A^{FEM}x = b$ can be approximated by

$$\tilde{x} := -\sum_{j=-k}^{k} \frac{2w_j}{\lambda_{min}} \bigotimes_{i=1}^{d} \exp\left(\frac{2t_j}{\lambda_{min}} M^{-1}A_i\right) M^{-1}b_i \tag{16}$$

with approximation error

$$\|x - \tilde{x}\| \leq \frac{C_{\mathrm{st}}}{\pi\lambda_{min}} \exp(\frac{2\mu\lambda_{min}^{-1}+1}{\pi} - \pi\sqrt{2k}) \oint_{\Gamma} \left\|(\lambda I - 2A^{FEM}/\lambda_{min})^{-1}\right\| \mathrm{d}_{\Gamma}\lambda \;\; \|\bigotimes_{i=1}^{d} M^{-1}b_i\|.$$

Proof: Instead of $Ax = b$ we consider the scaled equation $(2A/\lambda_{min})x = 2b/\lambda_{min}$. The matrix $2A/\lambda_{min}$ fulfils the requirements of Lemma 5 with $\Lambda = 2\lambda_{max}/\lambda_{min}$ and $2\mu/\lambda_{min}$ instead of μ. Application of Lemmata 2 and 3 yields the error estimates for the approximants (14) and (16). ■

Remark 7 *The relative error can be estimated by means of $\|b\| \leq \|A\| \, \|x\|$. This does not destroy the exponential decay of the quadrature error with respect to the rank k. In the finite element case (6), this reads*

$$\|\bigotimes_{i=1}^{d} M^{-1}b_i\| = \|\left(\bigotimes_{i=1}^{d} M^{-1}\right) \; A^{FEM} \; x\| = \|\widetilde{A}^{FEM} \; x\| \leq \|\widetilde{A}^{FEM}\|\|x\|.$$

In Lemma 6 we exploited the fact that the right-hand side is a tensor vector. In general this is not necessary - only the computation of the solution, i.e., the evaluation of the inverse is more complex. The approximate inverse (for arbitrary right-hand sides) can be represented in the tensor form of the next theorem.

Theorem 8 (Approximate Inverse) *Let A be a matrix of the tensor structure (2) with spectrum $\sigma(A)$ contained in the strip $\Omega := -[\lambda_{min}, \lambda_{max}] \oplus i[-\mu, \mu] \subseteq \mathbb{C}_-$. Let $k \in \mathbb{N}$ and t_j, w_j denote the points and weights from Lemma 4. Then the inverse A^{-1} to A can be approximated by*

$$\widetilde{A^{-1}} := -\sum_{j=-k}^{k} \frac{2w_j}{\lambda_{min}} \bigotimes_{i=1}^{d} \exp(\frac{2t_j}{\lambda_{min}} A_i) \tag{17}$$

with approximation error

$$\|A^{-1} - \widetilde{A^{-1}}\| \leq \frac{C_{\mathrm{st}}\|A\|}{\pi\lambda_{min}} \exp(\frac{2\mu\lambda_{min}^{-1}+1}{\pi} - \pi\sqrt{2k}) \oint_{\Gamma} \left\|(\lambda I - \frac{2}{\lambda_{min}} A)^{-1}\right\| \mathrm{d}_{\Gamma}\lambda.$$

Let A^{FEM} be a matrix of the tensor structure (6) and let the sum of the spectra of the $M^{-1}A_i$ be contained in Ω. Then the inverse to A^{FEM} can be approximated by

$$\widetilde{A^{-1}} := -\sum_{j=-k}^{k} \frac{2w_j}{\lambda_{min}} \bigotimes_{i=1}^{d} \exp(\frac{2t_j}{\lambda_{min}} M_i^{-1}A_i) M_i^{-1} \tag{18}$$

with approximation error

$$\|(A^{FEM})^{-1} - \widetilde{A^{-1}}\| \leq \frac{C_{\mathrm{st}}\|\widetilde{A}^{FEM}\|}{\pi\lambda_{min}} \exp(\frac{2\mu\lambda_{min}^{-1}+1}{\pi} - \pi\sqrt{2k}) \oint_{\Gamma} \left\|(\lambda I - \frac{2}{\lambda_{min}} A^{FEM})^{-1}\right\| \mathrm{d}_{\Gamma}\lambda.$$

Proof: Consider the scaled equation $(2A/\lambda_{min})x = 2b/\lambda_{min}$ and apply Lemmata 2 and 3. In the FEM case, the inequality $\| \left(\bigotimes_{i=1}^{d} M^{-1} \right) b \| \leq \| \widetilde{A}^{FEM} \| \|x\|$ is used. ■

5 LINEAR DIFFERENTIAL EQUATIONS

For discretised parabolic differential equations we have to compute a solution $x(t)$ of the (ordinary) linear differential equation

$$\partial_t x(t) = Ax(t), \qquad x(0) = b.$$

The solution is

$$x(t) = \exp(tA)b$$

which can easily be computed if the matrix A is of the tensor form (2). The Kronecker rank of the solution $x(t)$ is the same as for the initial value b. If b is a tensor vector (1), then the solution $x(t)$ is

$$x(t) = \bigotimes_{i=1}^{d} \exp(tA_i)b_i.$$

Here, we need to compute d matrix exponentials while in the previous section we had to compute $d(2k+1)$ matrix exponentials for the quadrature points t_j. Also, we do not need any assumption concerning the spectrum of A, since we are only interested in the evaluation of the matrix exponential at a certain finite time t.

6 ASSUMPTIONS ON THE RIGHT-HAND SIDE

At the beginning we demanded the right-hand side to be of the tensor form (1). Of course, the right-hand side b could also be the sum of m vectors $b^{(1)}, \ldots, b^{(m)}$ which are each of the tensor form (1): the approximate inverse $\widetilde{A^{-1}}$ has to be computed once and can then be evaluated for multiple right-hand sides. This enables us to deal with two important classes of right-hand sides.

6.1 Sparse Right-Hand Sides

If the right-hand side is sparse in the sense that b has only $m \ll n^d$ nonzero entries, then b can trivially be decomposed into m tensor vectors $b^{(1)}, \ldots, b^{(m)}$. Also, a single direction j may be dense such that

$$b = \otimes_{i=1}^{d} b_i, \quad \text{all } b_i \text{ except } b_j \text{ are unit vectors .} \tag{19}$$

6.2 Smooth Right-Hand Sides

If the right-hand side b of the equation stems from the pointwise evaluation of some function

$$f : [0,1]^d \to \mathbb{R},$$

which is not necessarily given in tensor form but smooth in the sense

$$|\partial_j f| \;\leq\; C_f\, \gamma^j\, j! \qquad\qquad (j \in \mathbb{N}_0, \gamma \geq 0), \tag{20}$$

then one can use a d-dimensional interpolation scheme to obtain an approximation of f by the sum of k_{rhs}^d tensor functions

$$f_i : [0,1]^d \to \mathbb{R}, \quad f_i(x_1, \ldots, x_d) = \otimes_{j=1}^d f_i^{(j)}(x_j).$$

Each of the functions f_i allows for a fast solution (their discretisation is a tensor vector of the form (1)) and the approximation error is estimated in the following Lemma.

Lemma 9 *Let $f : [0,1]^d \to \mathbb{R}$ be a smooth function in the sense of (20). We denote the one-dimensional Chebyshev interpolation points and weights by y_i, ω_i and the corresponding Lagrange polynomials by L_i. Then the function $\tilde{f} := \sum_{i \in \{0,\ldots,k_{rhs}\}^d} f_i$,*

$$f_i = f(y_{i_1}, \ldots, y_{i_d}) \prod_{j=1}^d \omega_{i_j} L_{i_j},$$

approximates f with an exponentially decaying error

$$|f(x) - \tilde{f}(x)| \leq 8e(2\log(k_{rhs}+1)/\pi)^d C_f (1+\gamma\sqrt{d})(1+k_{rhs})\left(\frac{\gamma\sqrt{d}}{2+\gamma\sqrt{d}}\right)^{k_{rhs}+1}.$$

Proof: Apply [1, Theorem 3.2] and exploit $\Lambda_{k_{rhs}} \leq 2\log(k_{rhs}+1)/\pi$ for the stability constant in the Chebyshev interpolation and $\operatorname{diam}([0,1]^d) \leq \sqrt{d}$. ∎

It should be noted that the dimension d enters the complexity for the solution in the exponent k_{rhs}^d such that really high-dimensional problems ($d > 10$) cannot be treated in this way. There, one has to study the right-hand side in more detail to exploit some kind of structure.

7 COMPUTATION

The representation formula (17) allows for a fast evaluation if the right-hand side of the equation $Ax = b$ is given in tensor form (1): we have to perform the matrix-vector multiplication of an $n \times n$ matrix $(2k+1)d$ times and the approximate solution $\tilde{x}$ is stored as the sum of tensor vectors. The computation of the $n \times n$ matrices $\exp(\frac{2t_j}{\lambda_{min}} A_i)$ requires the knowledge of the smallest eigenvalue λ_{min} of A. Since the eigenvalues of A are the sum of the eigenvalues of the A_i,

$$\sigma(A) \quad = \quad \sum_{i=1}^d \sigma(A_i) \quad = \quad \left\{ \sum_{i=1}^d \lambda_i \;\middle|\; \lambda_i \in \sigma(A_i) \right\},$$

it suffices to compute the smallest eigenvalue of each A_i. These can be obtained, e.g., by an inverse iteration.

For the computation of the matrix exponential there are quite a lot and different methods (see [10] for an overview). Two of them are of interest here and will be discussed in the next two subsections.

7.1 Diagonalisation

If we have obtained a decomposition of the matrix A_i,

$$A_i = T_i D_i T_i^{-1},$$

with diagonal matrix D_i that contains the eigenvalues and regular matrix T, that contains the eigenvectors of A_i, then we can compute the matrix exponential for different values of t_j by

$$\exp(\frac{2t_j}{\lambda_{min}} A_i) = T_i \exp(\frac{2t_j}{\lambda_{min}} D_i) T_i^{-1}.$$

The matrix exponential resolves into n scalar expressions. In the same way, we can treat block-diagonal matrices D_i.

Algorithm 10 (Computation by Diagonalisation)
Input: the matrices A_i and the tensor vector $\otimes_{i=1}^d b_i$. All A_i are diagonalisable.
Output: the approximate tensor solution $\tilde{x} = \sum_{j=-k}^{k} \otimes_{i=1}^d \tilde{x}_i^j$.

1. *Compute for each A_i the decomposition $A_i = T_i D_i T_i^{-1}$ with diagonal matrix D_i.*
2. *Transform the right-hand side $\hat{b}_i := T_i^{-1} b_i$.*
3. *Compute for each $1 \le i \le d$ and each $-k \le j \le k$ the vector*

$$\tilde{x}_i^j := \frac{2w_j}{\lambda_{min}} T_i \exp(\frac{2t_j}{\lambda_{min}} D_i) \hat{b}_i,$$

where $\lambda_{min} := \sum_{i=1}^d \lambda_{min}(A_i)$ and t_j, w_j from Lemma 4.

The advantage of this approach is that we can compute the eigenvector basis once and use it for all $2k+1$ quadrature points t_j. Moreover, the (up to machine precision) exact minimal eigenvalues of each A_i are known.

The drawback is that the complexity of the eigenvalue problem is cubic in the size of n such that the overall complexity for Algorithm 10 is $\mathcal{O}(dn^3 + (2k+1)dn^2)$. Moreover, the eigenvector system T_i may be severely ill-conditioned such that the numerical realisation becomes instable.

The conclusion is that this method is suitable if the matrices A_i are symmetric and n small. In the case $A_i = A_0$ for all $1 \le i \le d$, the complexity even reduces to $\mathcal{O}(n^3 + (2k+1)dn^2)$.

7.2 Hierarchical matrix representation

In this section we want to prove that the matrix exponential can be approximated in the hierarchical matrix format introduced by Hackbusch [9], at least for the interesting one-dimensional case. A more general existence result is given in [3] but here the proof can be greatly simplified. In the practical computations we use a simple algorithm based on the Taylor series expansion that we explain at the end of this section.

The hierarchical matrix format is based on the subdivision of a matrix into smaller subblocks, where each subblock is of low rank. A suitable data-sparse representation of a matrix of rank at most k is the R(k_e)-matrix format defined next.

Definition 11 (R(k_e)-matrix) *Let $k_e \in \mathbb{N}_0$. A matrix $M \in \mathbb{R}^{n\times m}$ is said to be given in R(k_e)-matrix representation if it is given in factorised form*

$$M = UV^T, \qquad U \in \mathbb{R}^{n\times k_e}, V \in \mathbb{R}^{m\times k_e}.$$

Definition 12 (Hierarchical matrix) *We define the hierarchical matrix ($\mathcal{H}$-matrix) format recursively. Let $k_e \in \mathbb{N}$. A matrix $M \in \mathbb{R}^{n\times n}$ is said to be given in $\mathcal{H}$-matrix format, if*

- $n \leq \max\{1, 2k_e\}$ *or*
- *M consists of four submatrices $M_{11}, M_{12}, M_{21}, M_{22}$ where M_{12}, M_{21} are R(k_e)-matrices and M_{11}, M_{22} are $\mathcal{H}$-matrices:*

$$M = \begin{bmatrix} M_{11} & M_{12} \\ M_{21} & M_{22} \end{bmatrix}.$$

The set of $\mathcal{H}$-matrices with blockwise rank k_e is denoted by $\mathcal{H}(k_e)$.

A typical hierarchical matrix is depicted in Figure 1. The subdivision should be so that M_{11} and

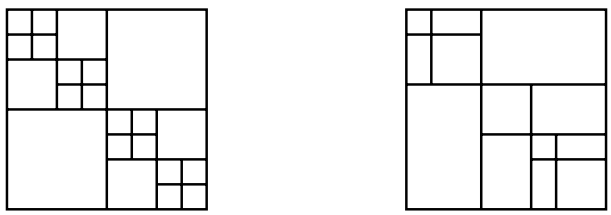

Figure 1: The empty squares represent R(k_e)-matrix blocks.

M_{22} are of almost equal size. Then the number of recursion steps is bounded by $\log(n)$.

The complexity to store and evaluate an $\mathcal{H}$-matrix is $\mathcal{O}(n\log(n)k_e)$ (see [9]). If we could approximate the matrix exponential $\exp(\frac{2t_j}{\lambda_{min}}A_i)$ by an $\mathcal{H}$-matrix with blockwise rank k_e, then the evaluation would be of complexity $\mathcal{O}((2k+1)dn\log(n)k_e)$ instead of $\mathcal{O}((2k+1)dn^2)$. Later we will observe that the matrix exponential can be computed with complexity $\mathcal{O}(n\log(n)^2k_e^2)$ such that the overall complexity is $\mathcal{O}((2k+1)dn\log(n)^2k_e^2)$ instead of $\mathcal{O}(dn^3 + (2k+1)dn^2)$ for the diagonalisation approach of the previous subsection.

More details concerning the $\mathcal{H}$-matrix arithmetic and the treatment of higher dimensional problems can be found in [6] and an introduction with applications is given in [2]. The proof of the following Lemma is contained in [9].

Lemma 13 *Let M be a tridiagonal regular matrix. Then the inverse M^{-1} is an $\mathcal{H}$-matrix with blockwise rank $k_e = 1$.*

The matrix exponential can be computed by discretisation of the Dunford-Cauchy integral formula (11). Since the integrand decays exponentially, it suffices to take logarithmically many quadrature points. The result from [3] is summarised in the following Lemma.

Lemma 14 *Let M be a matrix with spectrum contained in the strip $\Omega := -[\lambda_{min}, \lambda_{max}] \oplus i[-\mu, \mu] \subseteq \mathbb{C}_-$. Then the matrix exponential $\exp(M)$ can be approximated by a sum of resolvents*

$$\| \exp(M) - \sum_{j=-k_e}^{k_e} \kappa_j (z_j I - M)^{-1} \| \leq C \exp(4(\mu+1)^2 - (\mu+1)^{2/3} k_e^{2/3}).$$

Proof: The proof is given in [7, Lemma 4.6]. We use the fact that $\exp(M) = \exp(M - 2I + 2I) = \exp(M - 2I)e^2$ - then the spectrum of $M - 2I$ is contained in $-[2 + \lambda_{min}, 2 + \lambda_{max}] \oplus i[-\mu, \mu]$. ■

In the one-dimensional case the resolvents are all tridiagonal such that the inverses are of the $\mathcal{H}$-matrix format with blockwise rank 1. Since the approximation error in Lemma 14 decays with $k_e^{2/3}$ in the exponent, we need $k_e = \mathcal{O}(\log(\varepsilon)^{3/2})$ to achieve an accuracy of ε. A direct conclusion of Lemmata 13 and 14 is

Lemma 15 *Let M be a tridiagonal matrix with spectrum contained in the strip $\Omega := -[\lambda_{min}, \lambda_{max}] \oplus i[-\mu, \mu] \subseteq \mathbb{C}_-$. Then the matrix exponential $\exp(M)$ can be approximated by a matrix $M \in \mathcal{H}(2k_e + 1)$ with approximation error*

$$\| \exp(M) - E \| \leq C \exp(4(\mu+1)^2 - (\mu+1)^{2/3} k_e^{2/3}).$$

For the computation of the matrix exponential we use the Taylor-series approximation. This is a quite simple procedure where we replace the exact arithmetic (addition and multiplication) by the formatted $\mathcal{H}$-matrix arithmetic.

Algorithm 16 (Matrix exponential $\exp(tM)$)
The idea is to use the Taylor series representation $\exp(tM) = \sum_{\nu=0}^{\infty} M^\nu t^\nu / \nu!$ if the matrix fulfils $\|tM\| \leq 1/2$ and truncate the series due to exponential convergence. If $\|tM\| > 1/2$, then we first scale tM by 2^ℓ and square the result ℓ times:

1. *Compute an approximation to $\theta := \max\{\|tM\|, 1\}$, e.g., by power iteration.*
2. *Define $\ell := \lceil \log_2(\theta) \rceil$ and $\theta := 2^{-\ell}$.*
3. *Compute $E' := \sum_{\nu=0}^{10} M^\nu (t\theta)^\nu / \nu!$ and approximate E' by an $\mathcal{H}$-matrix $\tilde{E} \in \mathcal{H}(k_e)$.*

4. *Square the matrix* $\tilde{E}$ ℓ *times:* $E := (\tilde{E})^{2^\ell}$, *where the multiplication is performed by the formatted* $\mathcal{H}$*-matrix arithmetic.*

5. *Result:* $\exp(tM) \approx E \in \mathcal{H}(k_e)$.

In Algorithm 16 we computed the truncated Taylor series with 10 addends, because the remainder is then smaller than 10^{-9}. If instead 15 (20) addends are taken, then the remainder is smaller than 10^{-16} (10^{-24}).

Remark 17 (Choice of the rank k_e) *The rank k_e for the approximation of the matrix exponential in the set $\mathcal{H}(k_e)$ of hierarchical matrices should be taken according to the desired accuracy which is already limited by the accuracy ε of the quadrature formula with $2k+1$ quadrature points (the number of quadrature points is chosen such that the error bound (8) is smaller than ε). For a fixed number of quadrature points one can compare the rank k_e with a coarser approximation with bockwise rank $k_e - 1$ and take this as an indicator for the error $\delta_{ij} := \|\exp(\frac{2t_j}{\tilde{\lambda}_{min}} A_i) - E_{ij}\|$. For an even distribution of the error we demand*

$$\delta_{ij} < \lambda_{min}/((2k+1)2w_j).$$

Algorithm 18 (Computation by $\mathcal{H}$-matrix arithmetic)
Input: the matrices A_i and the tensor vector $\otimes_{i=1}^d b_i$.
Output: the approximate tensor solution $\tilde{x} = \sum_{j=-k}^{k} \otimes_{i=1}^d \tilde{x}_i^j$ of $Ax = b$.
For each $1 \le i \le d$ we compute

- *an approximation $\tilde{\lambda}_{min}(A_i)$ to the minimal real part of the eigenvalues of A_i (e.g., by inverse iteration).*

and the sum $\tilde{\lambda}_{min} := \sum_{i=1}^d \tilde{\lambda}_{min}(A_i)$. For each $1 \le i \le d$ and each $-k \le j \le k$ we compute

- *an approximation $E_{i,j} \in \mathbb{R}^{n \times n}$ to the matrix exponential $\exp(\frac{2t_j}{\tilde{\lambda}_{min}} A_i)$ by Algorithm 16 and the vector*

$$\tilde{x}_i^j := \frac{2w_j}{\tilde{\lambda}_{min}} E_{i,j} b_i$$

with t_j, w_j from Lemma 4.

8 NUMERICAL EXAMPLES

The numerical examples are restricted to finite difference discretisations on a tensor grid in the unit cube. At first we investigate the behaviour of our solution method with respect to the refinement of the discretisation and the increase of the dimension d for a symmetric problem. In the last part of this section we consider a convection dominated problem that gives rise to theoretical and practical difficulties.

Example 19 (Symmetric Model Problem) *Let $\Omega := [0,1]^d$ and $n \in \mathbb{N}$. We consider the differential equation*

$$\mathcal{A}u = f \quad \text{in } \Omega, \qquad u|_\Gamma = 0 \quad \text{on } \Gamma := \partial\Omega, \tag{21}$$

where the operator $\mathcal{A}$ is defined as

$$\mathcal{A}u := \sum_{i=1}^{d} \partial_i^2 u. \tag{22}$$

The right-hand side f for the equation is so that the solution is

$$u(x) = \prod_{i=1}^{d} 4(x_i - x_i^2), \tag{23}$$

i.e., f is the sum of $2d+1$ tensor functions. A standard finite difference discretisation of (21) on a uniform grid leads to the task of solving a linear system $Ax = b$ with a matrix A of the form (2) with tridiagonal matrices (cf. [8])

$$A_i = -\begin{bmatrix} 2h^{-2} & -h^{-2} & & \\ -h^{-2} & \ddots & \ddots & \\ & \ddots & \ddots & -h^{-2} \\ & & -h^{-2} & 2h^{-2} \end{bmatrix}. \tag{24}$$

The right-hand side b is a sum of $2d+1$ tensor vectors (1).

8.1 Low Dimension $d = 3$

In the case $d = 3$ we want to compare the result $\tilde{x}$ computed by our algorithm with the exact solution x of the equation $Ax = b$ and the corresponding function $\tilde{u}$ with the continuous solution u. The function u is contained in $C^\infty(\Omega)$ with vanishing third partial derivatives in each spatial direction. Therefore, the finite difference discretisation scheme yields a discrete solution $\tilde{u}$ that is in each gridpoint identical to the exact solution u, i.e., the pointwise discretisation error is zero such that the discrete solution of the system is the vector x with entries x_j equal to the value of u in the j-th gridpoint. From the knowledge of the continuous solution u we can represent the vector x in the tensor form (1).

We measure the error of the approximate solution $\tilde{x}$ in the Euclidean norm:

$$\varepsilon := \|\tilde{x} - x\|_2 / \|x\|_2.$$

The results for the three-dimensional case $d = 3$ with $k = 15$ in the quadrature rule and $n = 512, \ldots, 8192$ points per spatial direction ($N = n^3$ degrees of freedom) are contained in Table 1.

For small d the complexity is dominated by the number n of gridpoints per spatial direction. The $\mathcal{H}$-matrix arithmetic is advantageous for $n > 1000$ and since the complexity is linear in the dimension d one can immediately estimate the complexity for any d. Also, Table 1 resembles the fact that the error estimate (15) is independent of the fineness parameter n of the discretisation.

As a comparison we want to note that a tensor product multigrid method on this structured grid with $N = 1024^3$ degrees of freedom would take several hours to solve the problem while our new method solves this problem in a few minutes.

$N = n^3$ dof	t (Diagonalisation)	$\tilde{\varepsilon}$	t ($\mathcal{H}$-matrix)	$\tilde{\varepsilon}$
n=512	10	$3.0-6$	24	$3.0-6$
n=1024	110	$3.0-6$	79	$3.1-6$
n=2048	1573	$3.1-6$	247	$3.1-6$
n=4096	–	–	744	$3.2-6$
n=8192	–	–	2144	$3.1-6$

Table 1: Three-dimensional symmetric model problem: time in seconds for $k = 15$ in the quadrature rule. Accuracy estimated by random evaluation in 1000 entries.

8.2 High Dimension $d \gg 3$

Since the dimension d enters the complexity only linearly, we are almost independent of the dimension d of the underlying continuous problem. In order to demonstrate that the error estimate (15) is independent of the dimension we will give a numerical example.

We consider the model problem from Example 19 with $n := 1024$ and $d = 1, 2, 4, \ldots, 256$. The right-hand side has a Kronecker rank of $k_{rhs} = 2d + 1$ such that the complexity to compute the solution is quadratic in the dimension d. This limits the possible dimensions d, where we can compute and store the solution, to $d < 300$. The matrix exponentials are stored in the $\mathcal{H}$-matrix representation and computed in the formatted $\mathcal{H}$-matrix arithmetic. The numerical results from

$N = 1024^d$	time (seconds)	ε	$\tilde{\varepsilon}$
d=1	70	$3.8-6$	$3.8-6$
d=2	68	$2.2-6$	$2.2-6$
d=4	68	–	$3.0-6$
d=8	68	–	$2.4-6$
d=16	75	–	$2.2-6$
d=32	107	–	$2.0-6$
d=64	246	–	$1.6-6$
d=128	794	–	$3.3-6$
d=256	2981	–	$5.5-6$

Table 2: High-dimensional symmetric model problem: time in seconds for $n = 1024$, $k = 15$. Accuracy ε measured exactly (low dimension) and estimated ($\tilde{\varepsilon}$) by random evaluation in 1000 entries.

Table 2 confirm the independence of the approximation error from the dimension d.

8.3 Nonsymmetric Problem

In the previous section we considered an elliptic operator with real spectrum in the left complex halfplane. The discretisation led to a symmetric system matrix. In this section we consider a model problem with dominant convection, such that the spectrum is complex. In the error estimates for the approximate solution $\tilde{x}$ the absolute value of the complex parts enters in the exponent, but this

can be compensated by a higher rank k in the quadrature formula. Another obstruction is the term $\oint_\Gamma \left\|(\lambda I - 2A/\lambda_{min})^{-1}\right\| \mathrm{d}_\Gamma\lambda$. In the symmetric case we could bound $\|(\lambda I - 2A/\lambda_{min})^{-1}\|$ by 1 and the length of Γ by $2 + 4\lambda_{max}/\lambda_{min}$. In the non-symmetric case the value of $\|(\lambda I - 2A/\lambda_{min})^{-1}\|$ is not known and has to be compensated for by an increased rank.

Example 20 (Nonsymmetric Model Problem) *Let $\Omega := [0,1]^d$ and $n \in \mathbb{N}$. We consider the convection diffusion equation*

$$\mathcal{A}u = f \quad \text{in } \Omega, \qquad u|_\Gamma = 0 \quad \text{on } \Gamma := \partial\Omega, \tag{25}$$

where the operator $\mathcal{A}$ is defined as

$$\mathcal{A}u := \sum_{i=1}^{d} \partial_i^2 u - \sum_{i=1}^{d} c_i \partial_i u \tag{26}$$

with possibly dominant convection coefficients c_i. The right-hand side f for the equation is so that the solution is

$$u(x) = \prod_{i=1}^{d} 4(x_i - x_i^2). \tag{27}$$

We use a standard finite difference discretisation on a uniform grid for the diffusion term and a second order convergent scheme (Fromm's scheme) for the convection term. The discrete system matrix is of the form (2) with banded matrices

$$A_i = -\begin{bmatrix} 2h^{-2} + \frac{3}{4}c_i h^{-1} & -h^{-2} - \frac{5}{4}c_i h^{-1} & \frac{1}{4}c_i h^{-1} & \\ -h^{-2} + \frac{1}{4}c_i h^{-1} & 2h^{-2} + \frac{3}{4}c_i h^{-1} & -h^{-2} - \frac{5}{4}c_i h^{-1} & \frac{1}{4}c_i h^{-1} \\ \ddots & \ddots & \ddots & \ddots \\ & \ddots & \ddots & -h^{-2} + \frac{1}{4}c_i h^{-1} \\ & & -h^{-2} + \frac{1}{4}c_i h^{-1} & 2h^{-2} + \frac{3}{4}c_i h^{-1} \end{bmatrix}. \tag{28}$$

The right-hand side b is a sum of $2d+1$ tensor vectors (1).

As a first example we consider the parameter set $c_i = 100$, $n = 256$ and $d = 1$. The system matrix is the one from Example 20. The results for different values of k are presented in Table 3.

$c_i = 10^2$	$k = 15$	$k = 30$	$k = 60$	$k = 120$	$k = 240$
$\lvert x - \tilde{x}\rvert/\lvert x\rvert$	$7.9 - 1$	$5.8 - 1$	$2.4 - 1$	$5.7 - 3$	$3.6 - 8$

Table 3: Approximation error versus number k of quadrature points.

The approximation quality can be severely improved by choosing a "suitable" scaling factor: in Lemma 6 we scaled the equation $Ax = b$ by the factor $2|\lambda_{min}|^{-1}$ such that the maximal real part of the eigenvalues of $\frac{2}{|\lambda_{min}|}A$ is -2. Now, we scale the system by a factor of $\alpha|\lambda_{min}|^{-1} > 0$ where the parameter α has to be determined adaptively for the matrix A. The results for the example from Table 3 with the factor $\alpha := 3.0$ are contained in Table 4.

$c_i = 10^2$	$k = 15$	$k = 30$	$k = 60$	$k = 120$	$k = 240$
$\lvert x - \tilde{x}\rvert / \lvert x\rvert$	$4.8 - 4$	$1.3 - 5$	$6.8 - 8$	$2.4 - 11$	$4.8 - 14$

Table 4: Approximation error versus number k of quadrature points with additional shift $\alpha = 3.0$.

For each parameter α we can compute an approximation x_α to the solution x (fixed rank k) and measure the error

$$\varepsilon_\alpha := \|x - x_\alpha\| / \|x\|.$$

In the numerical examples it seems that the function $\alpha \mapsto \varepsilon_\alpha$ has a unique minimiser α and is moreover convex. The idea now is to exploit this and determine an (almost) optimal scaling factor α. To do this, we minimise the error with respect to a known solution x and a fixed number k of quadrature points. The (almost) optimal scaling factor can then be used for an arbitrary right-hand side, where the solution is not known.

For the one-dimensional minimisation problem we use a standard bisection strategy. The improvement can clearly be seen in Table 5, where we compare the by an optimal α scaled system with the unscaled one.

$c_i = 10^4$	$k = 15$	$k = 30$	$k = 60$	$k = 120$	$k = 240$
α	1	1	1	1	1
ε_1	$1.4 - 1$	$8.9 - 2$	$5.1 - 2$	$2.5 - 2$	$6.9 - 3$
α	0.54	0.42	0.37	0.44	0.5
ε_α	$6.5 - 2$	$2.0 - 2$	$2.8 - 4$	$1.7 - 6$	$1.2 - 11$

Table 5: Approximation error ε_α versus number k of quadrature points with shift $\alpha = 1$ in the second row and optimal α in the last row.

We close this section with a three-dimensional example where the convection coefficients are $c_1 = 100, c_2 = 1000, c_3 = 10000$ and the discretisation parameter is $n = 256$. The results in Table 6 show that it is possible to approximate the solution with a moderate number k of quadrature points.

$c = (10^2, 10^3, 10^4)$	$k = 15$	$k = 30$	$k = 60$	$k = 90$
α	1	1	1	1
ε_1	$7.9 - 2$	$4.3 - 2$	$1.8 - 2$	$8.5 - 3$
α	0.85	0.69	0.60	0.63
ε_α	$6.2 - 2$	$1.9 - 2$	$2.8 - 4$	$3.5 - 6$

Table 6: Approximation error ε_α versus number k of quadrature points with shift $\alpha = 1$ in the second row and optimal α in the last row.

9 CONCLUSIONS

We have presented a method for the approximate solution of a linear system where the system matrix is of the tensor structure arising typically from finite element and finite difference discretisations of a partial differential equation on a tensor grid. The inverse stiffness matrix can be approximated in a data sparse format as the sum of matrices in tensor structure. The complexity for the approximation of the inverse is almost linear with respect to the meshwidth h^{-1} and linear in the dimension d of the space where the partial differential equation is posed.

If the right-hand side is the sum of few tensor vectors, then an approximation to the solution of the system can be computed in $\mathcal{O}(dh^{-1}\log(h^{-1}))$.

References

[1] S. Börm, L. Grasedyck: Low-rank approximation of integral operators by interpolation. *Preprint* No. **72** (2002), Max-Planck-Institute for Mathematics in the Sciences, Leipzig.

[2] S. Börm, L. Grasedyck, W. Hackbusch. Introduction to Hierarchical Matrices with Applications. *Preprint* No. **18** (2002), Max-Planck-Institute for Mathematics in the Sciences, Leipzig.

[3] I. Gavrilyuk, W. Hackbusch, B. Khoromskij. $\mathcal{H}$-matrix approximation for the operator exponential with applications. *Numer. Math.* 2002; **92**:83–111.

[4] L. Grasedyck, W. Hackbusch. A Multigrid Method to Solve Large Scale Sylvester Equations. *Preprint*, Max-Planck-Institute for Mathematics in the Sciences, Leipzig. In preparation.

[5] L. Grasedyck. Existence of a low rank or $\mathcal{H}$-matrix approximant to the solution of a Sylvester equation. *Preprint* No. **2** (2002), University Kiel.

[6] L. Grasedyck, W. Hackbusch. Construction and arithmetics of hierarchical matrices. *Computing*; to appear.

[7] L. Grasedyck, W. Hackbusch, B. Khoromskij. Solution of large scale algebraic matrix Riccati equations by use of hierarchical matrices. *Computing* 2003; **70**:121–165.

[8] Hackbusch W. Elliptic Differential Equations. Theory and Numerical Treatment. *Springer*, New York, 1992; **62**:89–108.

[9] Hackbusch W. A sparse matrix arithmetic based on $\mathcal{H}$-matrices. Part I: Introduction to $\mathcal{H}$-matrices. *Computing* 2000; **62**:89–108.

[10] C. Moler, C. Van Loan. Nineteen dubious ways to compute the exponential of a matrix. *SIAM* Rev. 1978; **20**:801–836.

[11] T. Penzl. A cyclic low rank Smith method for large sparse Lyapunov equations. *SIAM* J. Sci. Comput. 2000; 21 (4):1401–1418.

[12] F. Stenger: Numerical methods based on Sinc and analytic functions. *Springer*, New York, 1993.

[13] V. Strassen: Gaussian elimination is not optimal. *Num. Math.* 1969; **13**:354–356.