A PARALLEL RANK-ADAPTIVE INTEGRATOR FOR DYNAMICAL LOW-RANK APPROXIMATION[*]

GIANLUCA CERUTI[†], JONAS KUSCH[‡], AND CHRISTIAN LUBICH[§]

Abstract. This work introduces a parallel and rank-adaptive matrix integrator for dynamical low-rank approximation. The method is related to the previously proposed rank-adaptive basis update & Galerkin (BUG) integrator but differs significantly in that all arising differential equations, both for the basis *and* the Galerkin coefficients, are solved in parallel. Moreover, this approach eliminates the need for a potentially costly coefficient update with augmented basis matrices. The integrator also incorporates a new step rejection strategy that enhances the robustness of both the parallel integrator and the BUG integrator. By construction, the parallel integrator inherits the robust error bound of the BUG and projector-splitting integrators. Comparisons of the parallel and BUG integrators are presented by a series of numerical experiments which demonstrate the efficiency of the proposed method, for problems from radiative transfer and radiation therapy.

Key words. Dynamical low-rank approximation, rank adaptivity, time integration

MSC codes. 68Q25, 68R10, 68U05

1. Introduction.

Dynamical low-rank approximation (DLRA) [13] has shown to be an efficient approach to reduce the computational cost and memory footprint of solving large matrix differential equations

$$\dot{\mathbf{A}}(t) = \mathbf{F}(t, \mathbf{A}(t))\,. \tag{1.1}$$

The core building block of dynamical low-rank approximation is to evolve an approximate solution on the low-rank manifold $\mathcal{M}$ by orthogonally projecting the vector-field $\mathbf{F}(t, \mathbf{A}(t))$ onto the tangent space of $\mathcal{M}$ at each instance in time. This approach is known as the time-dependent variational principle for the approximation manifold $\mathcal{M}$ in the physical and chemical literature, going back to Dirac [5] in the context of a quantum-mechanical problem that is approximated by separation of variables.

This approach evolves entirely on the low-rank manifold and thus differs from schemes which leave the manifold and then retract the numerical approximation back to the manifold. However, a severe challenge to the use of numerical time marching schemes is posed by high curvature of the manifold along the solution. The curvature is proportional to the inverse of the smallest nonzero singular value of the numerical approximation and the exact solution [13, §4]. The smallest nonzero singular value is typically very small, because the accuracy of a low-rank approximation corresponds to the largest discarded singular value of the solution, and then the smallest retained singular value is usually not much larger. The smallest nonzero singular value is therefore the smaller the higher the rank and the higher the accuracy requirements.

Overcoming the challenge posed by small singular values in constructing robust numerical schemes for dynamical low-rank approximation has been a major difficulty.

[*] Submitted to the editors April 12, 2023.

Funding: The work of J. Kusch was funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) - 491976834, and the work of Ch. Lubich was supported by DFG via FOR 5413 - 465199066 and SFB-TRR 352 - 470903074. The work of G. Ceruti was supported by the Swiss National Science Foundation (SNSF), grant number 200020-178806.

[†] Institute of Mathematics, EPF Lausanne, 1015 Lausanne, Switzerland.

[‡] Numerical Analysis and Scientific Computing, Universität Innsbruck, Technikerstraße 13, 6020 Innsbruck.

[§] Mathematisches Institut, Universität Tübingen, Auf der Morgenstelle 10, 72076 Tübingen, Germany.

So far, two numerical integrators have been developed that resolve this issue: the projector-splitting integrator [15] and the basis update & Galerkin (BUG) integrator [4]. The projector-splitting integrator applies a splitting scheme to the orthogonal projection onto the tangent space, which is an alternating sum of subprojections; it has been proved that this integrator is not affected by the presence of small singular values [12]. The BUG integrator, along with its rank-adaptive extension [2], updates the basis matrices similarly to the projector-splitting integrator but differs by performing the basis update in parallel while not requiring a backward-in-time step, which is important in strongly dissipative problems. The robustness of the BUG integrator is achieved through a perturbation argument applied to properties of the projector-splitting integrator. The basis augmentation step proposed in [2] allows for rank-adaptivity while exhibiting remarkable preservation properties.

Here we address some issues that remain with the rank-adaptive BUG integrator: First, while the basis update can be computed in parallel, the Galerkin step must be computed in a subsequent serial step. Second, the Galerkin update of double dimension requires a potentially expensive projection of the vector field onto the subspace spanned by the old *and* new bases. Third, the error due to the tangent space projection, which depends on the chosen rank, is not controlled in [2].

Already in [4, §3.1], it has been noted that BUG integrators offer the potential to compute the basis update *and* the Galerkin step in parallel. However, the modification given there lacks the robustness of the serial Galerkin step, as it requires a matrix inverse that is not under control except for step-sizes proportional to the smallest singular value. Another attempt to achieve full parallelism is found in the context of quantum physics [18], where the projector-splitting integrator is employed. The time-evolution of matrix product states is investigated and a computational algorithm using the message passing interface (MPI) architecture is achieved by recursively halving the matrix product state. Although this approach offers parallel features, it also suffers from small singular values [18, §III.D]. A different approach exploits parallelism in time as proposed in [1], which combines parallel-in-time methods [8] with dynamical low-rank approximation techniques.

In this work, we introduce a novel rank-adaptive integrator which updates the bases *and* performs the Galerkin update in parallel. Unlike the rank-adaptive BUG integrator [2], it does not require a Galerkin update with the augmented basis, which reduces computational costs. The new parallel integrator still shares the robust error bound of the projector-splitting and BUG integrators while no longer fulfilling the exactness property. The construction of the integrator is based on identifying new and old basis information in the rank-adaptive BUG integrator while using the fact that neglecting serially computed contributions still preserves the asymptotic error of the original integrator. Moreover, a step rejection strategy is proposed that can be used to efficiently control the low-rank error. This strategy applies to the parallel as well as the rank-adaptive BUG integrator of [2].

It should be noted that while the new parallel integrator is presented here for the matrix case of dynamical low-rank approximation, it is expected that its concepts extend to tensor trains and general tree tensor networks, where a much more substantial reduction of computation times by parallelism is to be gained. This will be explored in detail in future work, which will be based on ideas in the notationally and technically less demanding matrix case presented here.

The paper is structured as follows: In Section 2, we recapitulate the BUG integrator of [4] and its rank-adaptive formulation [2]. In Section 3, the new fully parallel integrator together with its rationale is introduced; a new step rejection strategy is discussed. In Section 4, the error analysis of the parallel integrator is presented. In Section 5, numerical experiments with examples from radiative transfer and radiation therapy validate the theoretical results of the new fully parallel integrator.

2. Recap: the rank-adaptive BUG integrator of [2].

The rank-adaptive parallel integrator presented in this paper has conceptual similarities to the Basis Update & Galerkin (BUG) integrator of [4] and its rank-adaptive version [2]. The rank-adaptive BUG integrator has been shown to be robust to small singular values in the exact solution $\mathbf{A}(t)$ or its low-rank approximation, and it has good conservation properties for Hamiltonian systems and dissipation properties for gradient systems up to the truncation tolerance ϑ. To understand how the new parallel integrator relates to and differs from these previously proposed integrators, it is useful to recall the rank-adaptive BUG integrator of [2]. Furthermore, in Section 3.3 we introduce criteria for step rejection, which are useful for the rank-adaptive integrator of [2] and play an even more important role for the parallel integrator of this paper.

2.1. Formulation of the rank-adaptive BUG integrator.

A time step of integration from time t_0 to $t_1 = t_0 + h$ starts from a factored rank-r_0 matrix $\mathbf{Y}_0 = \mathbf{U}_0\mathbf{S}_0\mathbf{V}_0^\top$ and computes an updated factorization $\mathbf{Y}_1 = \mathbf{U}_1\mathbf{S}_1\mathbf{V}_1^\top$ of rank $r_1 \le 2r_0$. We let $r = r_0$ and we put a hat on quantities related to the augmented rank $\widehat{r}$ with $r \le \widehat{r} \le 2r$ (and typically $\widehat{r} = 2r$). Let ϑ be a given rank truncation tolerance, just as a local error tolerance is provided by the user of an integrator with adaptive stepsizes.

1. Compute augmented basis matrices $\widehat{\mathbf{U}} \in \mathbb{R}^{m\times\widehat{r}}$ and $\widehat{\mathbf{V}} \in \mathbb{R}^{n\times\widehat{r}}$ (in parallel):

   K-step: Integrate from $t = t_0$ to t_1 the $m \times r$ matrix differential equation

   $$\dot{\mathbf{K}}(t) = \mathbf{F}(t, \mathbf{K}(t)\mathbf{V}_0^\top)\mathbf{V}_0, \qquad \mathbf{K}(t_0) = \mathbf{U}_0\mathbf{S}_0. \tag{2.1}$$

   Determine the columns of $\widehat{\mathbf{U}} \in \mathbb{R}^{m\times\widehat{r}}$ with $\widehat{r} \le 2r$ as an orthonormal basis of the range of the $m \times 2r$ matrix $(\mathbf{K}(t_1), \mathbf{U}_0)$ (e.g. by QR decomposition) and compute the $\widehat{r} \times r$ matrix $\widehat{\mathbf{M}} = \widehat{\mathbf{U}}^\top\mathbf{U}_0$.

   L-step: Integrate from $t = t_0$ to t_1 the $n \times r$ matrix differential equation

   $$\dot{\mathbf{L}}(t) = \mathbf{F}(t, \mathbf{U}_0\mathbf{L}(t)^\top)^\top\mathbf{U}_0, \qquad \mathbf{L}(t_0) = \mathbf{V}_0\mathbf{S}_0^\top. \tag{2.2}$$

   Determine the columns of $\widehat{\mathbf{V}} \in \mathbb{R}^{n\times\widehat{r}}$ as an orthonormal basis of the range of the $n \times 2r$ matrix $(\mathbf{L}(t_1), \mathbf{V}_0)$ (e.g. by QR decomposition) and compute the $\widehat{r} \times r$ matrix $\widehat{\mathbf{N}} = \widehat{\mathbf{V}}^\top\mathbf{V}_0$.

2. **Augment and update $\mathbf{S}_0 \to \widehat{\mathbf{S}}(t_1)$:**

   S-step: Integrate from $t = t_0$ to t_1 the $\widehat{r} \times \widehat{r}$ matrix differential equation

   $$\dot{\widehat{\mathbf{S}}}(t) = \widehat{\mathbf{U}}^\top\mathbf{F}(t, \widehat{\mathbf{U}}\widehat{\mathbf{S}}(t)\widehat{\mathbf{V}}^\top)\widehat{\mathbf{V}}, \qquad \widehat{\mathbf{S}}(t_0) = \widehat{\mathbf{M}}\mathbf{S}_0\widehat{\mathbf{N}}^\top. \tag{2.3}$$

3. **Truncate**: Compute the SVD $\widehat{\mathbf{S}}(t_1) = \widehat{\mathbf{P}}\widehat{\mathbf{\Sigma}}\widehat{\mathbf{Q}}^\top$ with $\widehat{\mathbf{\Sigma}} = \mathrm{diag}(\sigma_j)$ and truncate to the tolerance ϑ: Choose the new rank $r_1 \le \widehat{r} \le 2r$ as the minimal number r_1 such that

   $$\left(\sum_{j=r_1+1}^{\widehat{r}} \sigma_j^2\right)^{1/2} \le \vartheta.$$

   Compute the new factors for the approximation of $\mathbf{Y}(t_1)$ as follows: Let $\mathbf{S}_1$ be the $r_1 \times r_1$ diagonal matrix with the r_1 largest singular values and let $\mathbf{P}_1 \in \mathbb{R}^{\widehat{r}\times r_1}$ and $\mathbf{Q}_1 \in \mathbb{R}^{\widehat{r}\times r_1}$ contain the first r_1 columns of $\widehat{\mathbf{P}}$ and $\widehat{\mathbf{Q}}$, respectively. Finally, set $\mathbf{U}_1 = \widehat{\mathbf{U}}\mathbf{P}_1 \in \mathbb{R}^{m\times r_1}$ and $\mathbf{V}_1 = \widehat{\mathbf{V}}\mathbf{Q}_1 \in \mathbb{R}^{n\times r_1}$.

The approximation after one time step is given by

$$\mathbf{Y}_1 = \mathbf{U}_1\mathbf{S}_1\mathbf{V}_1^\top \approx \mathbf{Y}(t_1). \tag{2.4}$$

Then, $\mathbf{Y}_1$ is taken as the starting value for the next step, which computes $\mathbf{Y}_2$ in factorized form, etc.

The $m \times r$, $n \times r$ and $\widehat{r} \times \widehat{r}$ matrix differential equations in the substeps are solved approximately using a standard integrator, such as a Runge–Kutta method or an exponential integrator when $\mathbf{F}$ is predominantly linear.

To simplify the notation, we ignored that there might be different augmented ranks $\widehat{r}_U = \operatorname{rank}(\mathbf{K}(t_1), \mathbf{U}_0)$ and $\widehat{r}_V = \operatorname{rank}(\mathbf{L}(t_1), \mathbf{V}_0)$. This is readily taken into account in the above algorithm. In any case, typically one has $\widehat{r} = \widehat{r}_U = \widehat{r}_V = 2r$.

2.2. Interpretation. The K-step updates and augments the left basis from $\mathbf{U}_0$ to $\widehat{\mathbf{U}}$. The differential equation for $\mathbf{K}$ can be viewed as a Galerkin method in the subspace of matrices $\{\mathbf{K}\mathbf{V}_0^\top : \mathbf{K} \in \mathbb{R}^{m\times r}\} \subset \mathbb{R}^{m\times n}$ with fixed corange $\operatorname{Ran}(\mathbf{V}_0)$. Similarly, the differential equation for $\mathbf{L}$ is a Galerkin method in the subspace of matrices $\{\mathbf{U}_0\mathbf{L}^\top : \mathbf{L} \in \mathbb{R}^{n\times r}\} \subset \mathbb{R}^{m\times n}$ with fixed range $\operatorname{Ran}(\mathbf{U}_0)$.

The S-step is a Galerkin method for the differential equation (1.1) in the space of matrices $\widehat{\mathbf{U}}\mathbf{S}\widehat{\mathbf{V}}^\top$ generated by the extended basis matrices $\widehat{\mathbf{U}}$ and $\widehat{\mathbf{V}}$, with the projected starting value $\widehat{\mathbf{Y}}_0 = \widehat{\mathbf{U}}\widehat{\mathbf{U}}^\top\mathbf{Y}_0\widehat{\mathbf{V}}\widehat{\mathbf{V}}^\top = \widehat{\mathbf{U}}\widehat{\mathbf{S}}(t_0)\widehat{\mathbf{V}}^\top$. Since $\operatorname{Ran}\mathbf{U}_0 \subset \operatorname{Ran}\widehat{\mathbf{U}}$ and $\operatorname{Ran}\mathbf{V}_0 \subset \operatorname{Ran}\widehat{\mathbf{V}}$, the starting value is not changed: $\widehat{\mathbf{Y}}_0 = \mathbf{Y}_0 = \mathbf{U}_0\mathbf{S}_0\mathbf{V}_0^\top$.

3. The fully parallel integrator.

In the following, we present a novel parallel integrator that shares the robust error bound of the projector-splitting integrator [15] and of the Basis Update & Galerkin (BUG) integrator [4] as well as its rank-adaptive extension [2]. Important properties of the integrator are:

- All arising differential equations (for $\mathbf{K}$, $\mathbf{L}$ and $\mathbf{S}$) can be solved in parallel.
- All differential equations are evolved forward in time. This property is also valid for the BUG integrator, however not for the projector-splitting integrator, which requires a backward-in-time S-step.
- The integrator is rank-adaptive. That is, ranks are chosen in an automated way based on a user-determined tolerance parameter. Compared to the rank-adaptive BUG integrator, the new integrator does not require the sequential update of the $2r \times 2r$ augmented coefficient matrix $\widehat{S}$, which by itself is more costly than the parallel $r \times r$ S-step given below.

In contrast to the projector-splitting and the BUG integrators, the parallel integrator does not have the exactness property. That is, when the exact solution $\mathbf{A}(t)$ is of rank r and the right-hand side does not depend on the solution, i.e., $\mathbf{F} = \mathbf{F}(t) = \dot{\mathbf{A}}(t)$, the solution computed with the parallel integrator is in general not reproduced exactly. Moreover, unlike the rank-adaptive BUG integrator, the new integrator does not preserve the energy (up to a multiple of the truncation parameter) for Hamiltonian systems and does not necessarily diminish the energy (up to a multiple of the truncation parameter) for gradient systems.

3.1. Formulation of the algorithm.

A time step from t_0 to t_1 where the numerical solution at time t_0 is given by the factored rank-r matrix $\mathbf{Y}_0 = \mathbf{U}_0\mathbf{S}_0\mathbf{V}_0^\top$ reads:

1. Determine the augmented basis matrices $\widehat{\mathbf{U}} \in \mathbb{R}^{m\times 2r}$ and $\widehat{\mathbf{V}} \in \mathbb{R}^{n\times 2r}$ as well as the coefficient matrix $\bar{\mathbf{S}}(t_1) \in \mathbb{R}^{r\times r}$ (in parallel):

K-step: Integrate from $t = t_0$ to t_1 the $m \times r$ matrix differential equation

$$\dot{\mathbf{K}}(t) = \mathbf{F}(t, \mathbf{K}(t)\mathbf{V}_0^\top)\mathbf{V}_0, \qquad \mathbf{K}(t_0) = \mathbf{U}_0\mathbf{S}_0. \tag{3.1}$$

Determine $\widehat{\mathbf{U}} = (\mathbf{U}_0, \widetilde{\mathbf{U}}_1) \in \mathbb{R}^{m\times 2r}$ as an orthonormal basis of the range of the $m\times 2r$ matrix $(\mathbf{U}_0, \mathbf{K}(t_1))$ (e.g. by QR decomposition), where $\widetilde{\mathbf{U}}_1 \in \mathbb{R}^{m\times r}$ is filled with zero columns if $(\mathbf{U}_0, \mathbf{K}(t_1))$ has rank less than $2r$.
Compute the matrix $\widetilde{\mathbf{S}}_1^K = \widetilde{\mathbf{U}}_1^\top \mathbf{K}(t_1) \in \mathbb{R}^{r\times r}$.

L-step: Integrate from $t = t_0$ to t_1 the $n \times r$ matrix differential equation

$$\dot{\mathbf{L}}(t) = \mathbf{F}(t, \mathbf{U}_0\mathbf{L}(t)^\top)^\top\mathbf{U}_0, \qquad \mathbf{L}(t_0) = \mathbf{V}_0\mathbf{S}_0^\top. \tag{3.2}$$

Determine $\widehat{\mathbf{V}} = (\mathbf{V}_0, \widetilde{\mathbf{V}}_1) \in \mathbb{R}^{n\times 2r}$ as an orthonormal basis of the range of the $n \times 2r$ matrix $(\mathbf{V}_0, \mathbf{L}(t_1))$ (e.g. by QR decomposition), where $\widetilde{\mathbf{V}}_1 \in \mathbb{R}^{n\times r}$ is filled with zero columns if $(\mathbf{V}_0, \mathbf{L}(t_1))$ has rank less than $2r$.
Compute the matrix $\widetilde{\mathbf{S}}_1^L = \mathbf{L}(t_1)^\top \widetilde{\mathbf{V}}_1 \in \mathbb{R}^{r\times r}$.

S-step: Integrate from $t = t_0$ to t_1 the $r \times r$ matrix differential equation

$$\dot{\bar{\mathbf{S}}}(t) = \mathbf{U}_0^\top\mathbf{F}(t, \mathbf{U}_0\bar{\mathbf{S}}(t)\mathbf{V}_0^\top)\mathbf{V}_0, \qquad \bar{\mathbf{S}}(t_0) = \mathbf{S}_0. \tag{3.3}$$

2. **Augment**: Set up the augmented coefficient matrix $\widehat{\mathbf{S}}_1 \in \mathbb{R}^{2r\times 2r}$ as

$$\widehat{\mathbf{S}}_1 = \begin{pmatrix} \bar{\mathbf{S}}(t_1) & \widetilde{\mathbf{S}}_1^L \\ \widetilde{\mathbf{S}}_1^K & \mathbf{0} \end{pmatrix}. \tag{3.4}$$

3. **Truncate**: Compute the singular value decomposition $\widehat{\mathbf{S}}_1 = \widehat{\mathbf{P}}\widehat{\mathbf{\Sigma}}\widehat{\mathbf{Q}}^\top$ where $\widehat{\mathbf{\Sigma}} = \mathrm{diag}(\sigma_j)$. Truncate to the tolerance ϑ by choosing the new rank $r_1 \le 2r$ as the minimal number r_1 such that

$$\left(\sum_{j=r_1+1}^{2r} \sigma_j^2\right)^{1/2} \le \vartheta. \tag{3.5}$$

Compute the new factors for the approximation of $\mathbf{A}(t_1)$ as follows: Let $\mathbf{S}_1$ be the $r_1 \times r_1$ diagonal matrix with the r_1 largest singular values and let $\mathbf{P}_1 \in \mathbb{R}^{2r\times r_1}$ and $\mathbf{Q}_1 \in \mathbb{R}^{2r\times r_1}$ contain the first r_1 columns of $\widehat{\mathbf{P}}$ and $\widehat{\mathbf{Q}}$, respectively. Finally, set $\mathbf{U}_1 = \widehat{\mathbf{U}}\mathbf{P}_1 \in \mathbb{R}^{m\times r_1}$ and $\mathbf{V}_1 = \widehat{\mathbf{V}}\mathbf{Q}_1 \in \mathbb{R}^{n\times r_1}$.

The time-updated numerical solution is then obtained as $\mathbf{Y}_1 = \mathbf{U}_1\mathbf{S}_1\mathbf{V}_1^\top$.

3.2. Rationale for choosing the augmented matrix (3.4).

In the rank-adaptive BUG integrator described in Section 2, a Galerkin method is done in the space spanned by the augmented left and right bases $\widehat{\mathbf{U}}$ and $\widehat{\mathbf{V}}$. When these bases are chosen with the first r columns formed by $\mathbf{U}_0$ and $\mathbf{V}_0$, respectively (as is done above), then the differential equation (2.3) for $\widehat{\mathbf{S}}(t)$ becomes

$$\dot{\widehat{\mathbf{S}}}(t) = \widehat{\mathbf{U}}^\top\mathbf{F}(t, \widehat{\mathbf{U}}\widehat{\mathbf{S}}(t)\widehat{\mathbf{V}}^\top)\widehat{\mathbf{V}} = \begin{pmatrix} \mathbf{U}_0^\top\widehat{\mathbf{F}}(t)\mathbf{V}_0 & \mathbf{U}_0^\top\widehat{\mathbf{F}}(t)\widetilde{\mathbf{V}}_1 \\ \widetilde{\mathbf{U}}_1^\top\widehat{\mathbf{F}}(t)\mathbf{V}_0 & \widetilde{\mathbf{U}}_1^\top\widehat{\mathbf{F}}(t)\widetilde{\mathbf{V}}_1^\top \end{pmatrix} \tag{3.6}$$

with $\widehat{\mathbf{F}}(t) = \mathbf{F}(t, \widehat{\mathbf{U}}\widehat{\mathbf{S}}(t)\widehat{\mathbf{V}}^\top)$ for short, with the initial value

$$\widehat{\mathbf{S}}(t_0) = \begin{pmatrix} \mathbf{S}_0 & 0 \\ 0 & 0 \end{pmatrix}.$$

For the upper left block, this gives us, with $\bar{\mathbf{F}}(t) = \mathbf{F}(t, \bar{\mathbf{Y}}(t))$ for $\bar{\mathbf{Y}}(t) = \mathbf{U}_0 \bar{\mathbf{S}}(t) \mathbf{V}_0^\top$ and up to an $O(h^2)$ approximation error, under the assumption of a bounded and Lipschitz-continuous $\mathbf{F}$ as will be made in the following,

$$\widehat{\mathbf{S}}_{(0,0)}(t_1) = \mathbf{S}_0 + \int_{t_0}^{t_1} \mathbf{U}_0^\top \widehat{\mathbf{F}}(t) \mathbf{V}_0 \, dt \approx \mathbf{S}_0 + \int_{t_0}^{t_1} \mathbf{U}_0^\top \bar{\mathbf{F}}(t) \mathbf{V}_0 \, dt = \bar{\mathbf{S}}(t_1).$$

For the lower left block we have with $\mathbf{F}^K(t) = \mathbf{F}(t, \mathbf{K}(t)\mathbf{V}_0^\top)$ and again up to an $O(h^2)$ approximation error,

$$\begin{aligned} \widehat{\mathbf{S}}_{(1,0)}(t_1) &= \int_{t_0}^{t_1} \widetilde{\mathbf{U}}_1^\top \widehat{\mathbf{F}}(t) \mathbf{V}_0 \, dt \\ &\approx \int_{t_0}^{t_1} \widetilde{\mathbf{U}}_1^\top \mathbf{F}^K(t) \mathbf{V}_0 \, dt = \int_{t_0}^{t_1} \widetilde{\mathbf{U}}_1^\top \dot{\mathbf{K}}(t) \, dt = \widetilde{\mathbf{U}}_1^\top \mathbf{K}(t) = \mathbf{S}_1^K. \end{aligned}$$

Similarly we have up to $O(h^2)$

$$\widehat{\mathbf{S}}_{(0,1)}(t_1) \approx \widetilde{\mathbf{S}}_1^L.$$

Finally, the lower right block

$$\widehat{\mathbf{S}}_{(1,1)}(t_1) = \int_{t_0}^{t_1} \widetilde{\mathbf{U}}_1^\top \widehat{\mathbf{F}}(t) \widetilde{\mathbf{V}}_1 \, dt \tag{3.7}$$

is of size $O(h\varepsilon)$ if $\widehat{\mathbf{F}}(t_0) = \mathbf{F}(t_0, \mathbf{Y}_0)$ is tangential up to $O(\varepsilon)$ (with a small $\varepsilon > 0$) to the rank-r matrix manifold at $\mathbf{Y}_0 = \mathbf{U}_0 \mathbf{S}_0 \mathbf{V}_0$. This holds true because for tangential matrices $\mathbf{T}$ at $\mathbf{Y}_0$, which are of the form $\mathbf{T} = \mathbf{K}\mathbf{V}_0^\top + \mathbf{U}_0 \mathbf{L}^\top$ with arbitrary matrices $\mathbf{K}$ and $\mathbf{L}$ having r columns, we have $\widetilde{\mathbf{U}}_1^\top \mathbf{T} \widetilde{\mathbf{V}}_1^\top = 0$ as a consequence of the orthogonality relations $\widetilde{\mathbf{U}}_1^\top \mathbf{U}_0 = 0$ and $\widetilde{\mathbf{V}}_1^\top \mathbf{V}_0 = 0$. Approximate tangentiality to a low-rank matrix manifold of the vector field $\mathbf{F}(t, \mathbf{Y})$ near the solution of (1.1) is a condition that needs to be imposed for dynamical low-rank approximation, since the given vector field is projected onto the tangent space at the current approximation, at every instance of time.

Remark 3.1. An alternative rank-adaptive algorithm, which is closely related to the rank-adaptive BUG integrator, can be constructed by not setting the term $\widehat{\mathbf{S}}_{11}(t_1)$ to zero, but instead computing it in a serial step according to

$$\dot{\widehat{\mathbf{S}}}_{(1,1)} = \widetilde{\mathbf{U}}_1^\top \widehat{\mathbf{F}}(t) \widetilde{\mathbf{V}}_1, \quad \widehat{\mathbf{S}}_{(1,1)} = \mathbf{0} \in \mathbb{R}^{r \times r}, \tag{3.8}$$

or simply setting

$$\widehat{\mathbf{S}}_{11}(t_1) = h \, \widetilde{\mathbf{U}}_1^\top \mathbf{F}(t_0, \mathbf{Y}_0) \widetilde{\mathbf{V}}_1. \tag{3.9}$$

The core difference to the rank-adaptive BUG integrator is that this method does not require a $2r \times 2r$ coefficient update, but instead performs two $r \times r$ Galerkin steps (at least approximately). As we will see in the next subsection, the term in (3.9) is anyway computed to decide on whether the rank needs to be increased.

3.3. Step rejection strategy. To increase the robustness of the method, we propose to repeat the step in the following cases. This equally applies to the rank-adaptive BUG integrator of Section 2.

1. If $r_1 = 2r$, then repeat the step with the augmented left and right bases $\widehat{\mathbf{U}}$ and $\widehat{\mathbf{V}}$ in the role of $\mathbf{U}_0$ and $\mathbf{V}_0$, respectively.
2. If $r_1 < 2r$, compute $\eta = \|\widetilde{\mathbf{U}}_1^\top \mathbf{F}(t_0, \mathbf{Y}_0)\widetilde{\mathbf{V}}_1^\top\|$ and if $h\eta > c\vartheta$ (e.g., $c = 10$) with the tolerance ϑ used in (3.5), then repeat the step with the augmented left and right bases $\widehat{\mathbf{U}}$ and $\widehat{\mathbf{V}}$ in the role of $\mathbf{U}_0$ and $\mathbf{V}_0$.

ad 1. The proposed rank-adaptivity works remarkably well in many cases, as reported in [2] for the integrator of Section 2, but it has been noted in [11] that it does not work satisfactorily in cases where a steep increase of the rank is necessary within a step. This is because the rank cannot be more than doubled in a step, and this may not be sufficient, for example when starting from rank 1. A simple remedy is to repeat the step when no singular values are truncated, which indicates that an even higher rank is required. With this criterion, an arbitrary rank increase is possible from one step to the next, if needed.

ad 2. Good approximation can be expected only in the case of a small normal component of $\mathbf{F}(t, \mathbf{Y})$ with respect to the tangent space of the rank-r matrix manifold at $(t, \mathbf{Y})$; cf. [12]. More precisely, with a small $\varepsilon > 0$ we need

$$\|\mathrm{P}_r^\perp(\mathbf{Y}_0)\mathbf{F}(t_0, \mathbf{Y}_0)\| \le \varepsilon, \tag{3.10}$$

where $\mathrm{P}_r^\perp(\mathbf{Y}) = \mathrm{I} - \mathrm{P}_r(\mathbf{Y})$ and $\mathrm{P}_r(\mathbf{Y})$ is the orthogonal projection onto the tangent space $T_{\mathbf{Y}}\mathcal{M}_r$ of the rank-r matrix manifold $\mathcal{M}_r$ at $\mathbf{Y} \in \mathcal{M}_r$.

The algorithm as it stands does not control $\|\mathrm{P}_r^\perp(\mathbf{Y})\mathbf{F}(t, \mathbf{Y})\|$. While this quantity is not computationally accessible with a computational cost that is linear in $m + n$, we can inexpensively compute the normal component of the orthogonal projection $\widehat{\mathbf{U}}\widehat{\mathbf{U}}^\top \mathbf{F}_0 \widehat{\mathbf{V}}\widehat{\mathbf{V}}^\top$ of $\mathbf{F}_0 = \mathbf{F}(t_0, \mathbf{Y}_0)$ onto the augmented subspace spanned by $\widehat{\mathbf{U}}$ and $\widehat{\mathbf{V}}$, which is shown to be (by the proof below)

$$\|\mathrm{P}_r^\perp(\mathbf{Y}_0)[\widehat{\mathbf{U}}\widehat{\mathbf{U}}^\top \mathbf{F}_0 \widehat{\mathbf{V}}\widehat{\mathbf{V}}^\top]\| = \|\widetilde{\mathbf{U}}_1^\top \mathbf{F}_0 \widetilde{\mathbf{V}}_1\| = \eta. \tag{3.11}$$

Taking $\widehat{\mathbf{U}}\widehat{\mathbf{U}}^\top \mathbf{F}_0 \widehat{\mathbf{V}}\widehat{\mathbf{V}}^\top$ as a reduced model of $\mathbf{F}_0$ in the computationally inaccessible term $\|\mathrm{P}_r^\perp(\mathbf{Y}_0)\mathbf{F}_0\|$ and using (3.11) leads us to the step rejection criterion 2. This criterion is even more important for the new parallel integrator than for the integrator of Section 2 because the term $\widehat{S}_{(1,1)}$ of (3.7), which has norm $h\eta + O(h^2)$, is discarded in the algorithm.

Concerning the extra computational cost of computing η, we note that this is inexpensive if $\mathbf{F}_0$ is available (or is approximated) as a factorized low-rank matrix $\mathbf{F}_0 = \mathbf{G}\mathbf{H}^\top$ with slim matrices $\mathbf{G}$ and $\mathbf{H}$, since then $\widetilde{\mathbf{U}}_1^\top \mathbf{F}_0 \widetilde{\mathbf{V}}_1 = (\widetilde{\mathbf{U}}_1^\top \mathbf{G})(\widetilde{\mathbf{V}}_1^\top \mathbf{H})^\top$ is the product of two small matrices and the factors $\mathbf{G}$ and $\mathbf{H}$ need not be recomputed, but are still available from Step 1 of the integrator. If instead only matrix-vector products $\mathbf{F}_0\mathbf{v}$ are accessible, then $\mathbf{F}_0\widetilde{\mathbf{V}}_1$ must be computed anew. In this case, and unless the rank r is very small, it can suffice to roughly approximate η by taking only a few columns of $\widetilde{\mathbf{U}}_1$ and $\widetilde{\mathbf{V}}_1$ instead of all r columns.

Proof of (3.11). It is known from [13] that for $\mathbf{Y}_0 = \mathbf{U}_0\mathbf{S}_0\mathbf{V}_0^\top \in \mathcal{M}_r$ and $\mathbf{Z} \in \mathbb{R}^{m\times n}$,

$$\mathrm{P}_r^\perp(\mathbf{Y}_0)\mathbf{Z} = (\mathbf{I} - \mathbf{U}_0\mathbf{U}_0^\top)\mathbf{Z}(\mathbf{I} - \mathbf{V}_0\mathbf{V}_0^\top).$$

Here, $\mathbf{I} - \mathbf{U}_0\mathbf{U}_0^\top$ is the orthogonal projection onto the orthogonal complement of the range of $\mathbf{U}_0$. We choose a basis $\mathbf{U}_0^\perp$ of this space whose first r column vectors are those of $\widetilde{\mathbf{U}}_1$ and the remaining columns are orthogonal to $\widehat{\mathbf{U}} = (\mathbf{U}_0, \widetilde{\mathbf{U}}_1)$:

$$\mathbf{U}_0^\perp = (\widetilde{\mathbf{U}}_1, \widehat{\mathbf{U}}^\perp) \quad \text{with} \quad (\widehat{\mathbf{U}}^\perp)^\top \widehat{\mathbf{U}} = 0.$$

We then have $\mathbf{I} - \mathbf{U}_0\mathbf{U}_0^\top = \mathbf{U}_0^\perp(\mathbf{U}_0^\perp)^\top$ and hence

$$(\mathbf{I} - \mathbf{U}_0\mathbf{U}_0^\top)\widehat{\mathbf{U}}\widehat{\mathbf{U}}^\top = (\widetilde{\mathbf{U}}_1, \widehat{\mathbf{U}}^\perp)\begin{pmatrix} 0 & \mathbf{I} \\ 0 & 0 \end{pmatrix}\begin{pmatrix} \mathbf{U}_0^\top \\ \widetilde{\mathbf{U}}_1^\top \end{pmatrix} = \widetilde{\mathbf{U}}_1\widetilde{\mathbf{U}}_1^\top.$$

Analogously we have

$$(\mathbf{I} - \mathbf{V}_0\mathbf{V}_0^\top)\widehat{\mathbf{V}}\widehat{\mathbf{V}}^\top = \widetilde{\mathbf{V}}_1\widetilde{\mathbf{V}}_1^\top.$$

So we obtain

$$\begin{aligned} \mathrm{P}_r^\perp(\mathbf{Y}_0)[\widehat{\mathbf{U}}\widehat{\mathbf{U}}^\top\mathbf{F}_0\widehat{\mathbf{V}}\widehat{\mathbf{V}}^\top] &= (\mathbf{I} - \mathbf{U}_0\mathbf{U}_0^\top)\widehat{\mathbf{U}}\widehat{\mathbf{U}}^\top\mathbf{F}_0\big((\mathbf{I} - \mathbf{V}_0\mathbf{V}_0^\top)\widehat{\mathbf{V}}\widehat{\mathbf{V}}^\top\big)^\top \\ &= \widetilde{\mathbf{U}}_1\widetilde{\mathbf{U}}_1^\top\mathbf{F}_0\widetilde{\mathbf{V}}_1\widetilde{\mathbf{V}}_1^\top, \end{aligned}$$

which implies (3.11).

4. Robust error bound.

For the rank-adaptive BUG integrator described in Section 2, we derived in [2] local and global error bounds that are robust to small singular values, based on corresponding results in [12] and [4]. With the interpretation of the parallel integrator as a perturbed BUG integrator given in Section 3.2, such robust error bounds are also obtained for the parallel integrator.

4.1. Local error bound.

Let $\widehat{\mathbf{Y}}_1^{\mathrm{BUG}}$ be the rank-augmented numerical result of the BUG integrator of Section 2 after one step starting from $\mathbf{Y}_0$, and as above $\widehat{\mathbf{Y}}_1$ that of the new parallel integrator. Assuming that $\mathbf{F}$ is bounded with bounded derivatives, Section 3.2 shows that with the bound (3.10), the difference between these two numerical results is bounded by

$$\|\widehat{\mathbf{Y}}_1 - \widehat{\mathbf{Y}}_1^{\mathrm{BUG}}\| \le \widehat{c}_1 h^2 + \widehat{c}_2 h\varepsilon,$$

where $\widehat{c}_1$ and $\widehat{c}_2$ are independent of any singular values. It is readily verified that these constants actually only depend on a bound and a Lipschitz constant of $\mathbf{F}$. The local error bound for the rank-augmented BUG integrator given in [2] is of the same type if the initial value $\mathbf{A}(t_0)$ is of rank r and $\mathbf{Y}_0 = \mathbf{A}(t_0)$:

$$\|\widehat{\mathbf{Y}}_1^{\mathrm{BUG}} - \mathbf{A}(t_1)\| \le c_1^{\mathrm{BUG}} h^2 + c_2^{\mathrm{BUG}} h\varepsilon,$$

where again c_1 and c_2 only depend on a bound and a Lipschitz constant of $\mathbf{F}$. Furthermore, the rank-truncated matrix $\mathbf{Y}_1$ computed in the algorithm with truncation tolerance ϑ satisfies

$$\|\mathbf{Y}_1 - \widehat{\mathbf{Y}}_1\| \le \vartheta.$$

With the triangle inequality we thus obtain a robust local error bound for the parallel integrator:

$$\|\mathbf{Y}_1 - \mathbf{A}(t_1)\| \le c_1 h^2 + c_2 h\varepsilon + \vartheta, \tag{4.1}$$

where c_1 and c_2 only depend on a bound and a Lipschitz constant of $\mathbf{F}$.

4.2. Global error bound.

From the local error bound we conclude to a bound of the global error $\mathbf{Y}_n - \mathbf{A}(t_n)$ for $0 \le nh \le T$ with a fixed final time T using the familiar argument of Lady Windermere's fan [10, Section II.3] with error propagation by the exact flow. We do not give the details of the standard proof, but we formulate the resulting global error bound and begin with the assumptions:

1. $\mathbf{F}$ is bounded and Lipschitz continuous.

2. The normal component of $\mathbf{F}(t_n, \mathbf{Y}_n)$ is ε-small for $0 \le nh \le T$: With the rank r_n used in the nth time step,

$$\|\mathrm{P}_{r_n}^{\perp}(\mathbf{Y}_n)\mathbf{F}(t_n, \mathbf{Y}_n)\| \le \varepsilon.$$

(Note that the step rejection strategy of Section 3.3 aims at ensuring $h\varepsilon \le c\vartheta$.)

3. The error of the starting value $\mathbf{Y}_0$ of rank r_0 is δ-small: $\|\mathbf{Y}_0 - \mathbf{A}(0)\| \le \delta$.

We then have the following global error bound.

THEOREM 4.1. *Under assumptions* 1. *to* 3.*, the error of the parallel rank-adaptive integrator with stepsize h and rank-truncation tolerance ϑ is bounded by*

$$\|\mathbf{Y}_n - \mathbf{A}(t_n)\| \le C_0\delta + C_1 h + C_2\varepsilon + C_3\vartheta/h, \qquad 0 \le nh \le T,$$

where C_i $(i = 0, 1, 2, 3)$ only depend on the bound and Lipschitz constant of $\mathbf{F}$ and on the length T of the time interval.

5. Numerical results.

In the following, we provide numerical experiments to illustrate the behaviour of the new parallel integrator and compare it against the BUG integrator. The source code to reproduce all numerical experiments conducted in this manuscript is openly available and can be found in [3].

5.1. Planesource testcase of radiative transfer.

In the following, we investigate Ganapol's plane-source benchmark [7]. This test case considers an isotropic particle beam in slab geometry which is evolved in time by the radiative transfer equation. This equation models the transport and interaction of radiation particles with a background medium. For a particle density (or angular flux) $f = f(t, x, \mu)$, where $t \in \mathbb{R}_+$ denotes time, $x \in [a, b] \subset \mathbb{R}$ denotes space and $\mu \in [-1, 1]$ denotes the direction of flight (or angle), the radiation transport equation with isotropic scattering reads

$$\begin{aligned} &\partial_t f + \mu\partial_x f + f = \frac{1}{2}\int_{-1}^{1} f \, d\mu, \qquad (x, \mu) \in [a, b] \times [-1, 1], \\ &f(t = 0) = \frac{1}{3\sqrt{2\pi} \cdot 10^{-2}} \exp\Big(-\frac{x^2}{18 \cdot 10^{-4}}\Big). \end{aligned} \tag{5.1}$$

To discretize the solution f in space and angle, a spatial grid with N_x cells and midpoints $x_1 \le x_2 \le \cdots \le x_{N_x}$ with equidistant spacing Δx is chosen. In angle, a modal P_N approximation with N expansion coefficients (also known as moments) with respect to the normalized Legendre polynomials $p_i(\mu)$ is used. Then, in cell $I_j = [x_j - \Delta x/2, x_j + \Delta x/2]$ the entries of the solution matrix $\mathbf{Y} \in \mathbb{R}^{N_x \times N}$ are given as

$$Y_{ji}(t) \simeq \frac{1}{\Delta x}\int_{I_j}\int_{-1}^{1} f(t, x, \mu) p_{i-1}(\mu) \, \mathrm{d}\mu \mathrm{d}x. \tag{5.2}$$

An upwind discretization in space yields the matrix ODE

$$\dot{\mathbf{Y}} = -\mathbf{D}_x\mathbf{Y}\mathbf{A}^\top + \mathbf{D}_{xx}\mathbf{Y}|\mathbf{A}|^\top + \mathbf{Y}\mathbf{G} =: \mathbf{F}(\mathbf{Y}), \tag{5.3}$$

where $\mathbf{A} \in \mathbb{R}^{N \times N}$ is the flux matrix with entries $a_{\ell k} = \int_{-1}^{1} \mu p_\ell(\mu) p_k(\mu) \, \mathrm{d}\mu$ and $\mathbf{G} = \mathrm{diag}(0, 1, \cdots, 1) \in \mathbb{R}^{N \times N}$ is the scattering matrix. $\mathbf{A}$ is diagonalizable with real eigenvalues, i.e., $\mathbf{A} = \mathbf{T}\mathbf{\Lambda}\mathbf{T}^{-1}$ where $\mathbf{T}$ is a transformation matrix and $\mathbf{\Lambda}$ is a real-valued diagonal matrix. The absolute value of the flux matrix is then defined as $|\mathbf{A}| := \mathbf{T}|\mathbf{\Lambda}|\mathbf{T}^{-1}$. The tridiagonal stencil matrices are given as

$$D_{x,j,j\pm 1} = \pm\frac{1}{2\Delta x}, \quad D_{xx,j,j\pm 1} = \frac{\Delta x}{2h}, \quad \text{and} \quad D_{xx,jj} = \frac{\Delta x}{h}.$$

A forward Euler time discretization of (5.2) is stable under the time step restriction $h = \mathrm{CFL} \cdot \Delta x$, where $\mathrm{CFL} \in (0, 1]$. It is straightforward to show that the parallel integrator will inherit this CFL conditon: Note that the proposed integrator performs a Galerkin update with the basis

$$\mathbf{W} = [\mathbf{U}_0 \otimes \mathbf{V}_0, \widetilde{\mathbf{U}}_1 \otimes \mathbf{V}_0, \mathbf{U}_0 \otimes \widetilde{\mathbf{V}}_1] \in \mathbb{R}^{N_x \cdot N \times 3r^2}.$$

We vectorize the flux matrix $\mathbf{F}(\mathbf{Y})$ at $\mathbf{Y}_0$ as $\mathbf{f} \in \mathbb{R}^{N_x \times N}$ according to

$$f_{\mathrm{idx}(i,j)} = (\mathbf{F}(\mathbf{Y}_0))_{ij}\,, \quad \text{where } \mathrm{idx}(i,j) = (i-1) \cdot N_x + j$$

as well as the numerical solution $\mathbf{Y}_0$ as $\mathbf{y} \in \mathbb{R}^{N_x \times N}$ with entries $y_{\mathrm{idx}(i,j)} = (\mathbf{Y}_0)_{ij}$. Then, with the Euclidean norm $\| \cdot \|_E$, we have

$$\|\widehat{\mathbf{Y}}_1\| = \|\left(\mathbf{y} + h\mathbf{f}\right)\mathbf{W}\|_E \leq \|\mathbf{y} + h\mathbf{f}\|_E = \|\mathbf{Y}_0 + h\mathbf{F}(\mathbf{Y}_0)\|.$$

From the time step restriction of the full problem we have $\|\mathbf{Y}_0 + h\mathbf{F}(\mathbf{Y}_0)\| \leq \|\mathbf{Y}_0\|$ and the truncation fulfills $\|\mathbf{Y}_1\| \leq \|\widehat{\mathbf{Y}}_1\|$. Hence $\|\mathbf{Y}_1\| \leq \|\mathbf{Y}_0\|$.

We are interested in the so-called scalar flux $\Phi(t,x) = \int_{-1}^{1} f(t,x,\mu)\,d\mu$ which is given on the discrete level as $\mathbf{\Phi} \in \mathbb{R}^{N_x}$ with entries $\Phi_j = Y_{j1}$. We choose a spatial domain of $[-5, 5]$ which is discretized by $N_x = 1000$ spatial cells. The angular dependence is represented by $N = 500$ moments and the time step size is determined with $\mathrm{CFL} = 0.99$. As tolerance parameter we choose $\vartheta = \bar{\vartheta} \cdot \|\hat{\mathbf{\Sigma}}\|$, where $\bar{\vartheta} = 10^{-2}$. Moreover, the rejection step uses a value of $c = 1$ in the criterion $h\eta > c\vartheta$ and we use all r basis vectors in $\widetilde{U}_1$ and $\widetilde{V}_1$ to compute η.

The left column of Figure 1 depicts the scalar flux Φ at times $t \in \{2, 2.75, 5\}$ for the BUG and the new parallel integrator. The right column shows the rank evolution over time (top for BUG and bottom for the parallel integrator). It is observed that both integrators yield similar solution qualities as well as ranks. The resulting values for η as well as the criterion of the rejection step $c\vartheta/h$ are presented in Figure 2. We observe that the rejection step is not required in this testcase since the value of η is sufficiently small throughout the entire time interval. The parallel integrator at the coarse tolerance has a runtime of 26.8 seconds, whereas the rank-adaptive BUG integrator requires 47.6 seconds to run. At the fine tolerance, the parallel integrator computes the results in 39.4 seconds compared to 58.6 seconds for the rank-adaptive BUG integrator. That is, even at the fine tolerance, the parallel integrator is significantly faster than the rank-adaptive BUG integrator at the coarse tolerance.

5.2. Linesource testcase of radiative transfer. In the following, we extend the planesource testcase to a two-dimensional setting. This extension is known as the linesource testcase. For this benchmark, one approximates the solution $f = f(t, \mathbf{x}, \mathbf{\Omega})$ to the two-dimensional radiative transfer equation

$$\begin{aligned} &\partial_t f + \mathbf{\Omega} \cdot \nabla f + \sigma_t f = \frac{\sigma_s}{4\pi} \int_{\mathbb{S}^2} f \, d\mathbf{\Omega}, \qquad (\mathbf{x}, \mathbf{\Omega}) \in [-1.5, 1.5]^2 \times \mathbb{S}^2\,, \\ &f(t=0) = \max\left\{10^{-4}, \frac{1}{4\pi\sigma^2} \cdot \exp\left(-\frac{\|\mathbf{x}\|^2}{4\sigma^2}\right)\right\}. \end{aligned} \tag{5.4}$$

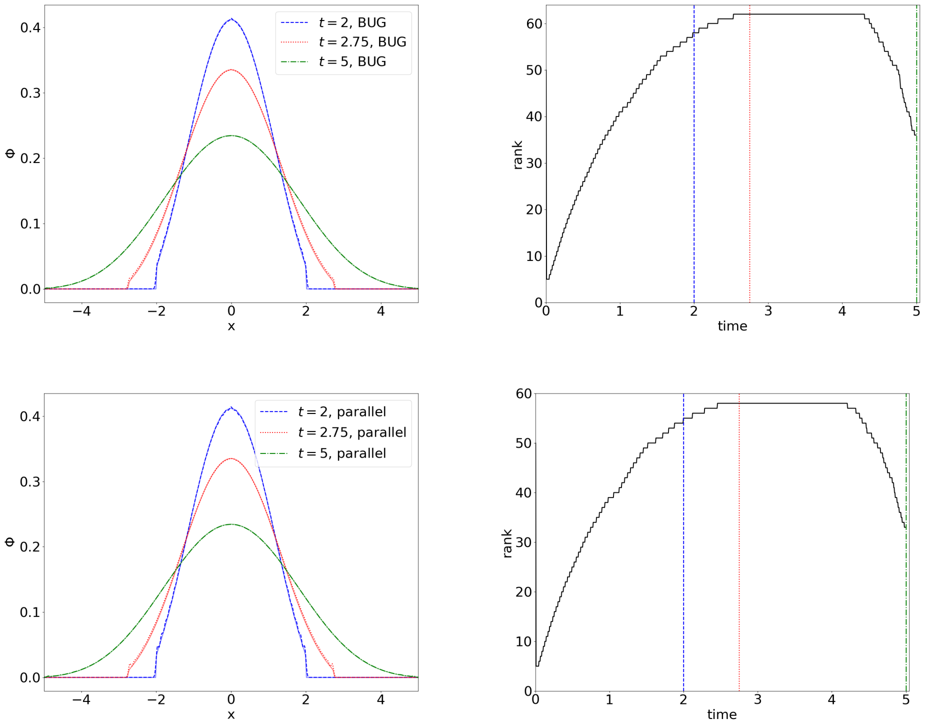


Fig. 1: Top: Rank-adaptive BUG integrator. Bottom: Parallel integrator. Left: Scalar flux of the reference solution (solid lines) of the radiation transport equation (5.1) in comparison with the approximation of the DLRA integrators at different times. Right: Rank evolution of the approximation arising from the rank-adaptive integrator.

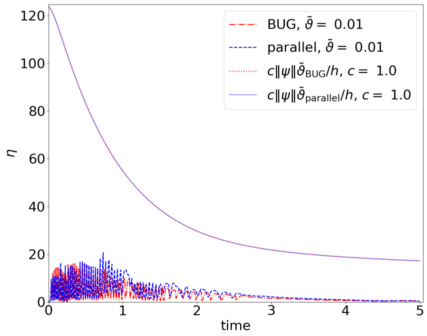
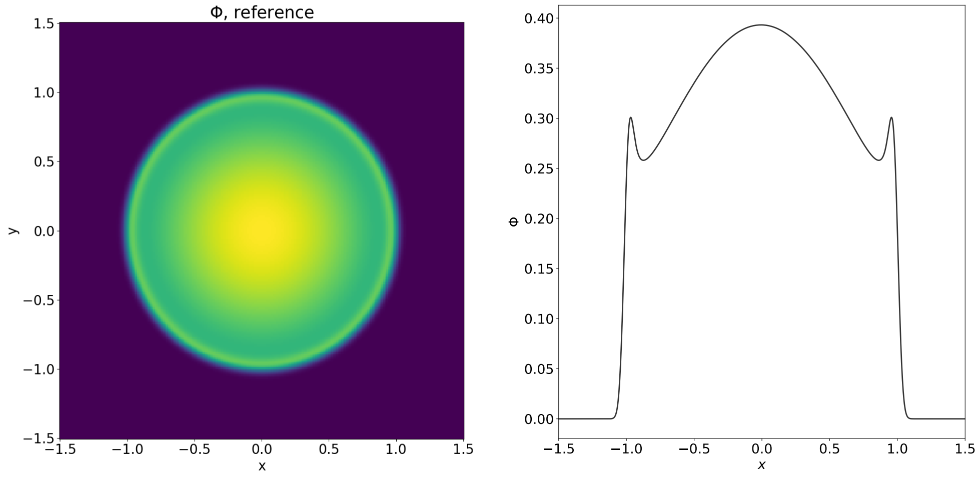


Fig. 2: Recorded values for η and rejection bound over time. It can be seen that even for a small value of $c = 1$ the time update is never rejected.



(a)

(b)

Fig. 3: Left: Scalar flux of reference solution. Right: Scalar flux of reference solution on a cut along $y = 0$.

Here, the two-dimensional spatial variable is denoted as $\mathbf{x} = (x, y)^\top$ and the angle (i.e., the direction of flight) is $\boldsymbol{\Omega} \in \mathbb{R}^3$. To model a constant solution along the z-axis, we set $\nabla = (\partial_x, \partial_y, 0)^\top$. The initial condition is a Gaussian pulse with standard deviation $\sigma = 0.03$ and an isotropic distribution in direction of flight. The background medium through which particles move has a scattering cross-section $\sigma_s = 1$. The total cross section is defined as $\sigma_t = \sigma_s + \sigma_a$, where $\sigma_a = 0$ denotes the absorption cross-section. This benchmark is remarkably challenging for conventional numerical methods as it exposes deficiencies of individual discretizations. For more information, see e.g., [9]. This testcase is also challenging for low-rank methods as it requires high ranks to yield good approximations, see e.g., [17, 14, 16]. At a final time $t_{\text{end}} = 1$, we are interested in the scalar flux $\Phi(t_{\text{end}}, \mathbf{x}) = \int_{\mathbb{S}^2} f(t, \mathbf{x}, \boldsymbol{\Omega})\, \mathrm{d}\boldsymbol{\Omega}$. The linesource testcase is equipped with an analytic solution [6, 7] which is depicted in Figure 3. To approximate the solution to the linesource testcase, we choose an equidistant spatial discretization of 250×250 spatial cells. The directional domain is discretized with a modal expansion in spherical harmonics basis functions. A polynomial degree of 39 which corresponds to 1600 expansion coefficients in angle is chosen. We use an upwind scheme with a forward Euler time integration to discretize space and time. As time step restriction, we use the CFL condition of the original problem and set $h = 0.7 \cdot \Delta x$, where $\Delta x = \Delta y$ is the cell width in x. The chosen constant in the bound for η is $c = 5$.

We observe that the ranks of the BUG as well as the parallel integrator will differ significantly when choosing the same tolerance parameter ϑ for both methods. Following [2], we pick tolerances $\bar{\vartheta} \in \{0.05, 0.025, 0.01\}$ for the BUG integrator as well as $\bar{\vartheta} \in \{0.005, 0.003, 0.0015\}$ for the parallel integrator, which yields comparable ranks. The resulting scalar fluxes are found in Figure 4. Figure 5 depicts the solution on a cut along $y = 0$. The chosen ranks are depicted in Figure 6 together with the resulting time evolution of η. We observe that both integrators yield similar numerical results at comparable ranks. Due to the fact that the parallel integrator does not necessi-

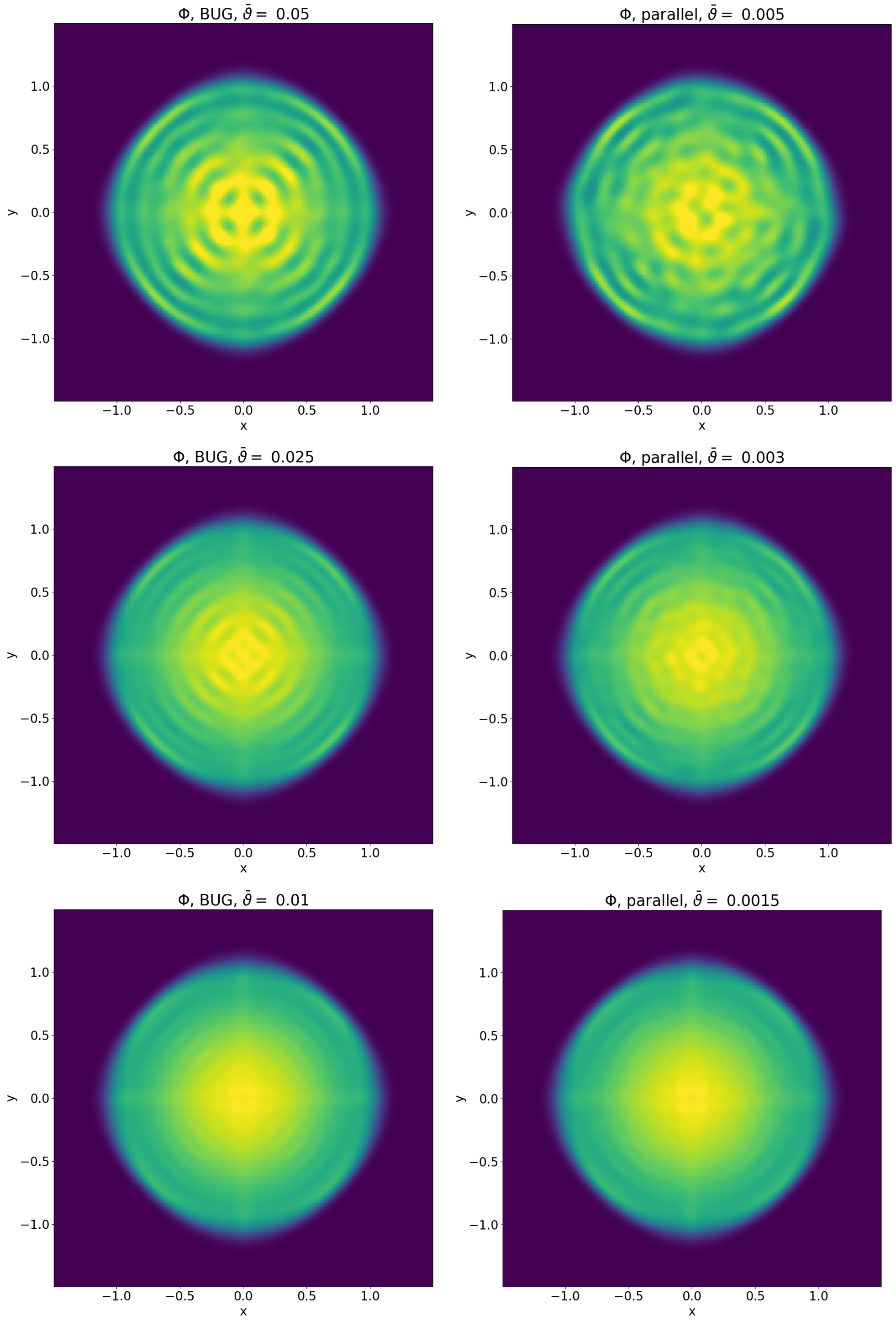


Fig. 4: Left: Adaptive BUG integrator. Right: Parallel integrator.

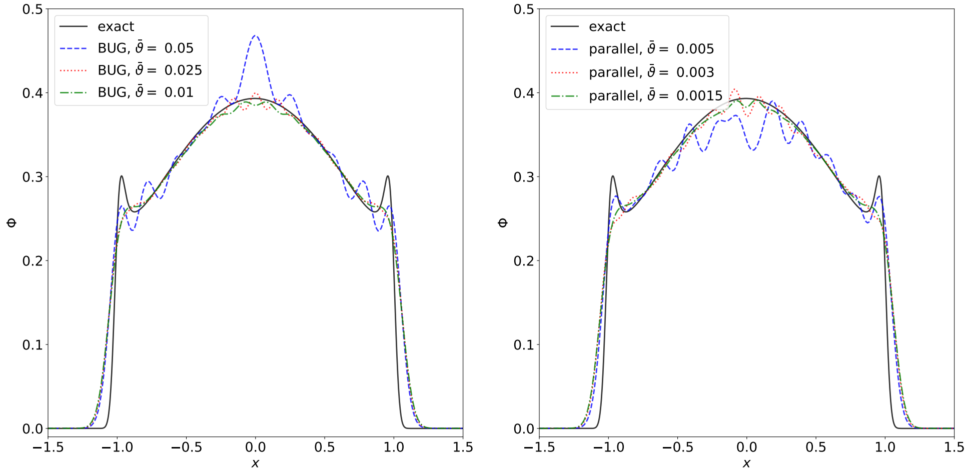


Fig. 5: Left: Adaptive BUG integrator. Right: Parallel integrator.

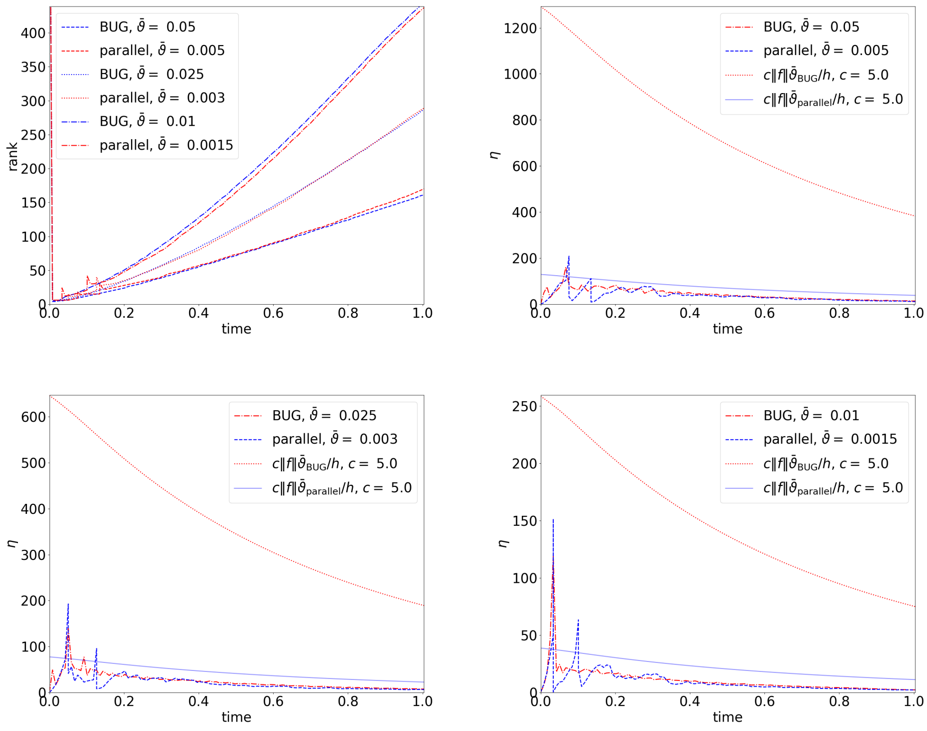


Fig. 6: Rank and η evolution in time.

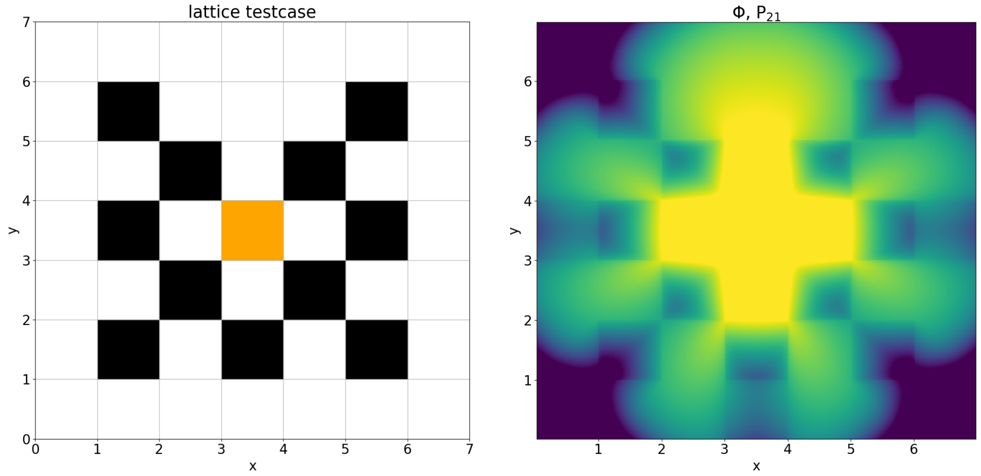
tate a computation of a $2r \times 2r$ coefficient matrix update, its computational time is substantially decreased. At the coarse tolerance, the parallel integrator requires 186 seconds compared to 256 seconds for BUG. At the intermediate tolerance, the parallel integrator requires 308 seconds compared to 462 seconds for BUG. At the fine tolerance, the parallel integrator requires 506 seconds compared to 782 seconds for BUG. When the problem is sufficiently big and justifies the communication overhead of a parallel implementation, the use of the full parallelism will further benefit the parallel integrator. Such an implementation is however left for future work on parallel tensor integrators.

Note that the recorded values for η depicted in Figure 9 indicate that a limited number of rejection steps are required for the parallel integrator in the beginning of the simulation. This is especially the case, since the chosen tolerance is significantly smaller compared to the tolerance of the BUG integrator.

5.3. Lattice testcase of radiative transfer. In this section, we investigate the lattice testcase, which mimics the blocks in a nuclear reactor. For this, we wish to determine the solution to the two-dimensional radiative transfer equation

$$\begin{aligned} &\partial_t f + \mathbf{\Omega} \cdot \nabla f + \sigma_t(\mathbf{x}) f = \frac{\sigma_s(\mathbf{x})}{4\pi} \int_{-1}^{1} f \, d\mathbf{\Omega} + Q(\mathbf{x}), \qquad (\mathbf{x}, \mathbf{\Omega}) \in [0,7]^2 \times \mathbb{S}^2 \,. \\ &f(t=0) = 10^{-9}. \end{aligned} \tag{5.5}$$

In contrast to previous numerical examples, the scattering and absorption cross-sections of the lattice testcase are spatially dependent. The spatial layout of the testcase can be found in Figure 7a. Here, white blocks indicate material which has only scattering properties, that is, $\sigma_s = 1$ and $\sigma_a = \sigma_t - \sigma_s = 0$. Black blocks are strongly absorbing with $\sigma_a = 10$ and no scattering, i.e., $\sigma_s = 0$. The orange block is an absorbing block with a source of $Q = 1$. A reference solution has been computed with the spherical harmonics (P_N) method where we use a polynomial degree of 21 which corresponds to 441 expansion coefficients in the direction of flight $\mathbf{\Omega}$. The chosen number of spatial cells is 350 in each spatial dimension, i.e., a total number of $350^2 = 122500$ cells is used. The scalar flux $\Phi(t, \mathbf{x}) = \int_{\mathbb{S}^2} f(t, \mathbf{x}, \mathbf{\Omega}) \, \mathrm{d}\mathbf{\Omega}$ is depicted in Figure 7b. As commonly done for the lattice benchmark, we us a logarithmic scale. In the following, we wish to approximate the P_{21} solution with rank-adaptive dynamical low-rank approximation integrators. We recall that the rank adaptive BUG integrator will pick increased ranks compared to the parallel integrator when both methods use the same tolerance parameter ϑ. Therefore, we run the BUG integrator with tolerances $\bar{\vartheta} = \{0.016, 0.01\}$. To obtain the same maximal rank, we run the parallel integrator with tolerances $\bar{\vartheta} = \{0.01, 0.005\}$. The corresponding scalar fluxes can be seen in Figure 8. It is observed that the parallel integrator and the BUG integrator yield similar results when the tolerance is chosen accordingly. However, the parallel integrator leads to negative values in certain regions which are depicted in white. This behaviour is not uncommon for modal approximations in radiative transfer and can also appear for the BUG integrator. The ranks chosen during the simulation, which can be found in Figure 9 are similar for both the BUG and the parallel integrator. Even though the parallel integrator increases the rank more slowly than BUG, the overall solution quality appears to not suffer from the reduced ranks except for regions with negative scalar fluxes. At the coarse tolerances ($\bar{\vartheta} = 0.016$ for BUG and $\bar{\vartheta} = 0.01$ for the parallel integrator) a maximal rank of 27 is picked, which leads to minor numerical artifacts in the scattering regions. At the fine tolerances



(a)

(b)

Fig. 7: Left: Setup lattice testcase. Right: P_{21} solution of lattice testcase.

($\bar{\vartheta} = 0.01$ for BUG and $\bar{\vartheta} = 0.005$ for the parallel integrator) a maximal rank of 39 is chosen. Here, the approximations agree well with the full P_{21} solution. In terms of computational cost, the runtime of the parallel integrator is 180 seconds for $\bar{\vartheta} = 0.01$ and 330 seconds for $\bar{\vartheta} = 0.005$. The BUG integrator has an increased runtime of 286 seconds for $\bar{\vartheta} = 0.016$ and 524 seconds for $\bar{\vartheta} = 0.01$. It is important to point out that the implementation of both method does not make use of the opportunity to compute K, L (and for the parallel integrator S) in parallel, which would further favor the parallel integrator. The improved runtime of the parallel integrator results mostly from omitting the $2r \times 2r$ coefficient update.

5.4. Electron therapy. In this section, we investigate the effectiveness of the parallel integrator for electron therapy. The main interest of computational radiation therapy is to predict the effectiveness of treatment plans by simulating the treatment procedure. The success of radiation treatment is determined by investigating the radiation dose absorbed by the patient at the tumor. At the same time, the radiation dose should not harm risk regions such as important organs or the spine. The movement of electrons through tissue is modeled by the continuous-slowing down equation

$$\begin{aligned}
\boldsymbol{\Omega} \cdot \nabla \psi(E, \mathbf{x}, \boldsymbol{\Omega}) + \rho(\mathbf{x}) \Sigma_t(E) \psi(E, \mathbf{x}, \boldsymbol{\Omega}) = &\int_{\mathbb{S}^2} \rho(\mathbf{x}) \Sigma_s(E, \boldsymbol{\Omega} \cdot \boldsymbol{\Omega}') \psi(E, \mathbf{x}, \boldsymbol{\Omega}')\, d\boldsymbol{\Omega}' \\
&+ \partial_E \left(\rho(\mathbf{x}) S_t(E) \psi(E, \mathbf{x}, \boldsymbol{\Omega}) \right) , \\
\psi(E, \mathbf{x}, \boldsymbol{\Omega}) = &\ \psi_{\mathrm{BC}}(E, \mathbf{x}, \boldsymbol{\Omega}) \quad \text{for } \mathbf{x} \in \partial D , \\
\psi(E_{\max}, \mathbf{x}, \boldsymbol{\Omega}) = &\ \psi_{\max}(\mathbf{x}, \boldsymbol{\Omega}) .
\end{aligned}$$

In contrast to previous examples, we are interested in computing a steady state solution. Additionally, the radiation flux ψ depends on the energy $E \in [0, E_{\max}]$. The tissue density $\rho(\mathbf{x})$ can be determined from a CT scan of the patient and the energy and angle-dependent cross sections Σ_t and Σ_s as well as the stopping power S_t can be determined from physical data bases. To construct an efficient numerical algorithm,

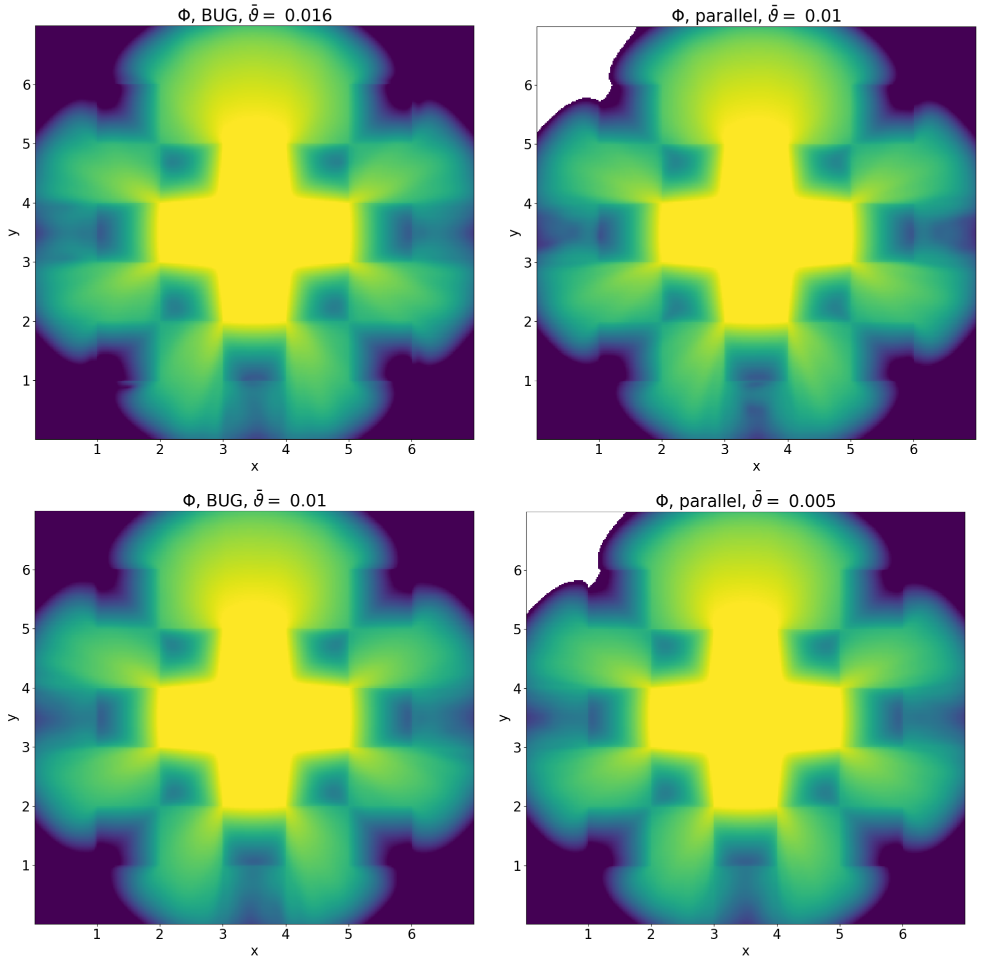


Fig. 8: Left: Adaptive BUG integrator. Right: Parallel integrator. Negative values are depicted in white.

the energy is transformed into a pseudo-time t according to

$$t(E) := \int_0^{E_{\max}} \frac{1}{S_t(E')}\, dE' - \int_0^{E} \frac{1}{S_t(E')}\, dE' \ . \tag{5.6}$$

Then, for the transformed radiation flux

$$f(t, \mathbf{x}, \mathbf{\Omega}) := S_t(E(t))\rho(\mathbf{x})\psi(E(t), \mathbf{x}, \mathbf{\Omega}) \tag{5.7}$$

we obtain the transformed continuous-slowing down equation

$$\partial_t f = -\mathbf{\Omega} \cdot \nabla_{\mathbf{x}} \frac{f}{\rho} - \widetilde{\Sigma}_t f + \int_{\mathbb{S}^2} \widetilde{\Sigma}_s(t, \mathbf{\Omega} \cdot \mathbf{\Omega}') f(t, \mathbf{x}, \mathbf{\Omega}')\, d\mathbf{\Omega}' \ , \tag{5.8}$$

where the tilde denotes a transformation of energy. We then evolve the solution in (pseudo) time t with the BUG and parallel integrators. For more details on the chosen

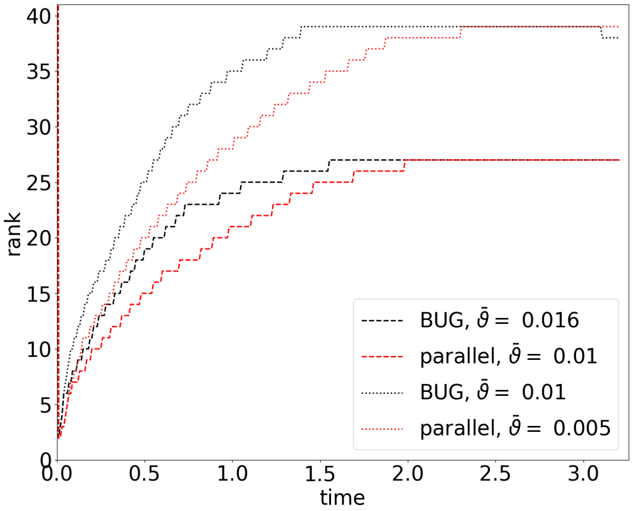


Fig. 9: Rank evolution in time.

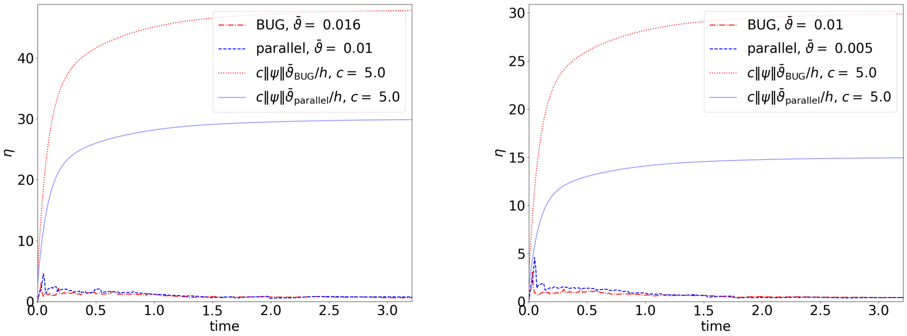


Fig. 10: Values of η in time.

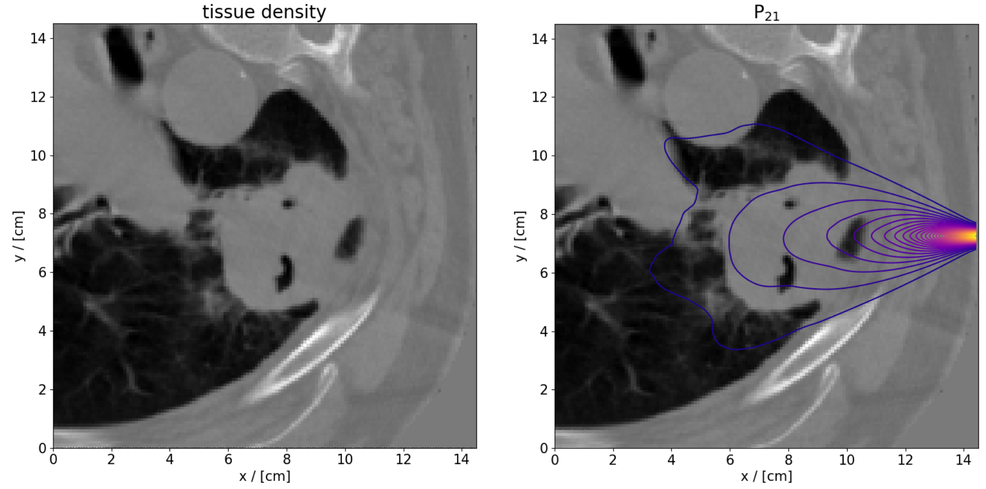
DLRA approximation we refer to [14]. To test the parallel integrator, we follow [14] and determine the dose

$$D(\mathbf{x}) = \frac{1}{\rho(\mathbf{x})} \int_0^\infty \int_{\mathbb{S}^2} S_t(E)\psi(E, \mathbf{x}, \mathbf{\Omega})\, d\mathbf{\Omega} dE \tag{5.9}$$

when radiating a lung tumor with an energy beam of strength $E_{\text{max}} = 21$ which is modelled by

$$\begin{aligned}\psi_{\text{in}}(E, \mathbf{x}, \mathbf{\Omega}) = 10^5 &\cdot \exp(-(\Omega_{1,*} - \Omega_1)^2/\sigma_{\Omega_1}) \cdot \exp(-(E_{\text{max}} - E)^2/\sigma_E) \\ &\cdot \exp(-(x_* - x)^2/\sigma_x) \cdot \exp(-(y_* - y)^2/\sigma_y)\ .\end{aligned}$$

Parameters of the ingoing beam are



(a)

(b)

Fig. 11: Left: Setup radiation therapy application. Right: Corresponding P$_{21}$ solution.

$x_* = 7.25, y_* = 14.5$	spatial mean of particle beam in cm
$\Omega_{1,*} = 1$	directional mean of particle beam
$\sigma_{\Omega_1}^{-1} = 75$	inverse directional beam variance
$\sigma_x^{-1} = \sigma_y^{-1} = 20$	inverse spatial beam variance
$\sigma_E^{-1} = 100$	inverse energy variance

We use the upwind method from previous examples with an equidistand spatial grid of 200 × 200 spatial cells. Again a modal discretization of angle is chosen, where we use spherical harmonics basis functions of degree up to 21. The tissue density row is depicted in Figure 11a and the corresponding full rank P$_{21}$ solution which is depicted in Figure 11b has been computed with a runtime of 17467 seconds. To reduce computational costs we approximate the dose distribution with DLRA using the BUG integrator with a tolerance of $\vartheta = 0.01$ and compare it to the parallel integrator with a tolerance of $\vartheta = 0.007$, which yields the comparable maximal rank. The corresponding doses are shown in Figure 12. We observe that both integrators yield satisfactory approximations which agree well with the P$_{21}$ reference solution. The runtimes are 5423 seconds for the BUG integrator and 3894 seconds for the parallel integrator. As expected, we observe fluctuations on the upper and lower left side, which are negligibly small. The corresponding ranks over energy can be found in Figure 13. Dominant spatial modes computed with BUG and the parallel integrator can be found in Figure 14 and dominant directional modes are depicted in Figure 15.

Conclusion. In the present contribution, a novel fully parallel numerical integrator for dynamical low-rank approximation has been presented. Through its construction as an approximation to the rank-adaptive BUG integrator, the new integrator

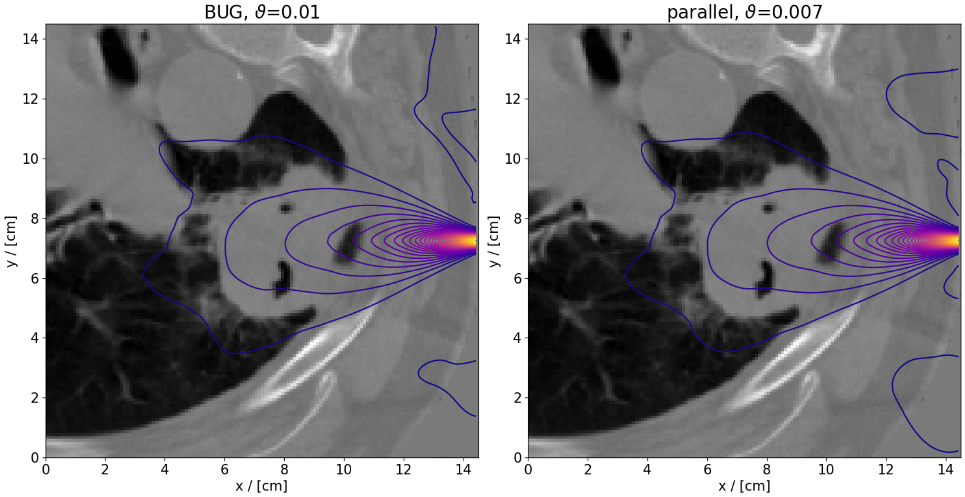


Fig. 12: Left: Adaptive BUG integrator. Right: Parallel integrator.

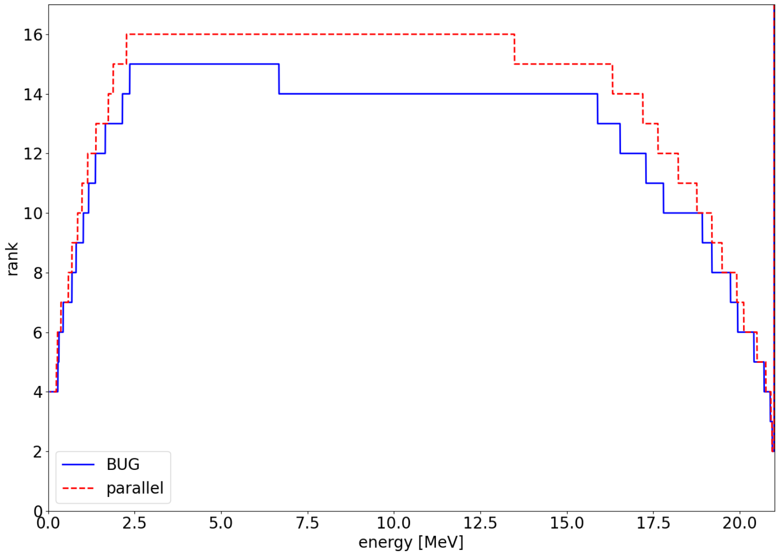


Fig. 13: Rank over energy

inherits the desirable robust error bound of both the BUG and the projector-splitting integrators. The full parallelism comes at the cost of no longer fulfilling the exactness property, which however does not impact the solution quality in the investigated numerical experiments. Beyond the enhanced parallelism, there is also a computational advantage in the serial setting, since the proposed integrator does not need a coefficient update with the augmented left and right basis matrices. The inherent parallelism of the new low-rank matrix integrator is expected to extend to yielding significant advantages for Tucker and tree tensor network formats, which we aim to

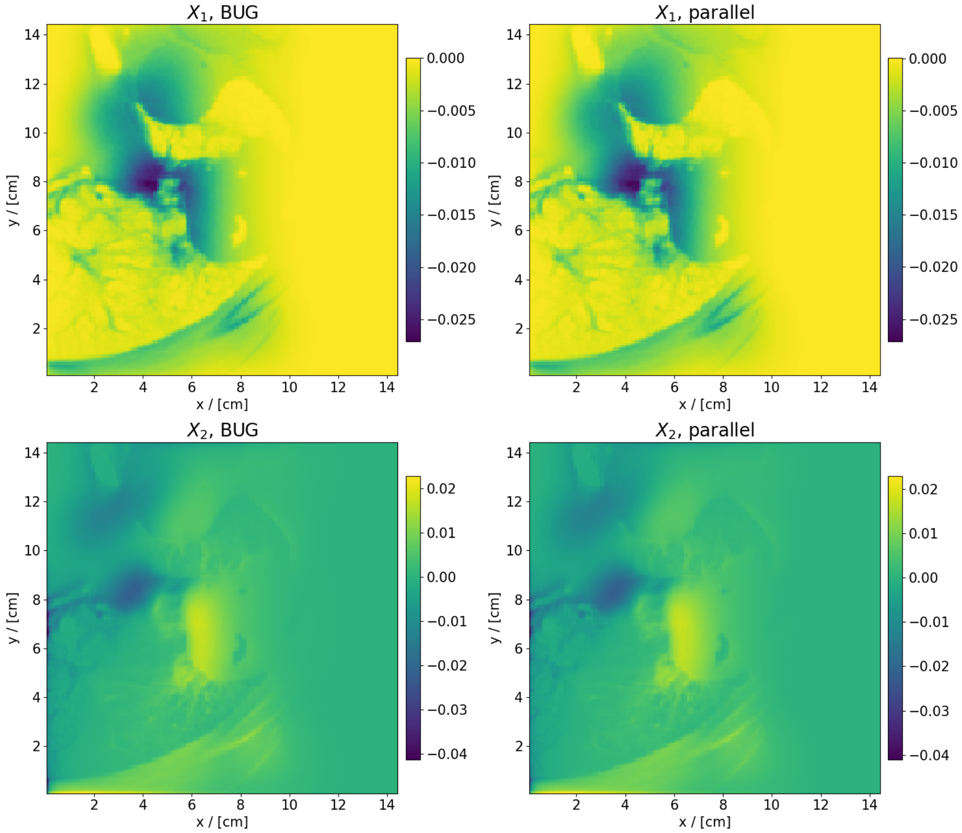


Fig. 14: First two dominant spatial modes X_1 and X_2 at $E = 0$.

explore in future work.

REFERENCES

[1] B. Carrel, M. J. Gander, and B. Vandereycken, *Low-rank Parareal: a low-rank parallel-in-time integrator*, BIT Numer. Math., 63 (2023), pp. Paper No. 13, 24.

[2] G. Ceruti, J. Kusch, and C. Lubich, *A rank-adaptive robust integrator for dynamical low-rank approximation*, BIT Numer. Math., 62 (2022), pp. 1149–1174.

[3] G. Ceruti, J. Kusch, and C. Lubich, *Numerical testcases for "A parallel rank-adaptive integrator for dynamical low rank approximation"*, 2023. https://github.com/JonasKu/Publication-A-parallel-rank-adaptive-integrator-for-dynamical-low-rank-approximation.git.

[4] G. Ceruti and C. Lubich, *An unconventional robust integrator for dynamical low-rank approximation*, BIT Numer. Math., 62 (2022), pp. 23–44.

[5] P. Dirac, *Note on exchange phenomena in the Thomas atom*, Proc. Cambridge Phil. Soc., 26 (1930), pp. 376–385.

[6] B. Ganapol, *Homogeneous infinite media time-dependent analytic benchmarks for x-tm transport methods development*, Los Alamos National Laboratory, (1999).

[7] B. Ganapol, *Analytical benchmarks for nuclear engineering applications*, Nuclear Energy Agency, (2008).

[8] M. J. Gander and S. Vandewalle, *Analysis of the parareal time-parallel time-integration method*, SIAM J. Sci. Comput., 29 (2007), pp. 556–578.

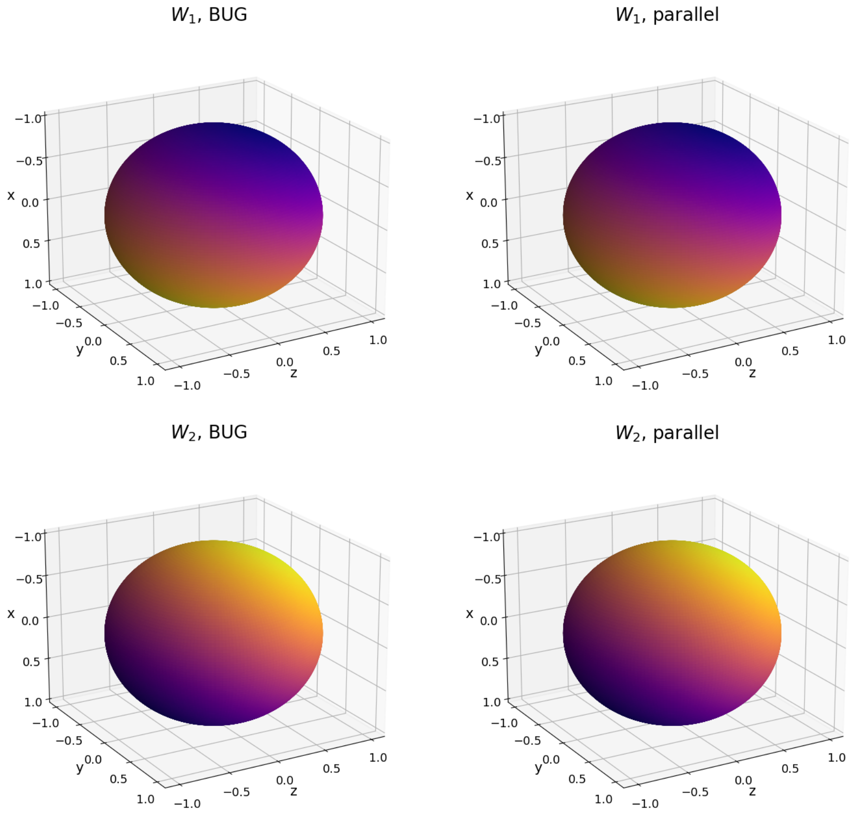


Fig. 15: First two dominant directional modes W_1 and W_2 at $E = 0$.

[9] C. K. Garrett and C. D. Hauck, *A comparison of moment closures for linear kinetic transport equations: The line source benchmark*, Transport Theory and Statistical Physics, 42 (2013), pp. 203–235.

[10] E. Hairer, S. P. Nørsett, and G. Wanner, *Solving Ordinary Differential Equations I. Nonstiff Problems*, Springer, Berlin, 2nd ed., 1993.

[11] C. Hauck and S. Schnake, *A predictor-corrector strategy for adaptivity in dynamical low-rank approximations*, arXiv preprint arXiv:2209.00550, (2022).

[12] E. Kieri, C. Lubich, and H. Walach, *Discretized dynamical low-rank approximation in the presence of small singular values*, SIAM J. Numer. Anal., 54 (2016), pp. 1020–1038.

[13] O. Koch and C. Lubich, *Dynamical low-rank approximation*, SIAM J. on Matrix Anal. Appl., 29 (2007), pp. 434–454.

[14] J. Kusch and P. Stammer, *A robust collision source method for rank adaptive dynamical low-rank approximation in radiation therapy*, arXiv preprint arXiv:2111.07160, (2021).

[15] C. Lubich and I. V. Oseledets, *A projector-splitting integrator for dynamical low-rank approximation*, BIT Numer. Math., 54 (2014), pp. 171–188.

[16] Z. Peng and R. G. McClarren, *A sweep-based low-rank method for the discrete ordinate transport equation*, J. Comput. Phys., 473 (2023), p. 111748.

[17] Z. Peng, R. G. McClarren, and M. Frank, *A low-rank method for two-dimensional time-dependent radiation transport calculations*, J. Comput. Phys., 421 (2020), p. 109735.

[18] P. Secular, N. Gourianov, M. Lubasch, S. Dolgov, S. R. Clark, and D. Jaksch, *Parallel time-dependent variational principle algorithm for matrix product states*, Phys. Rev. B, 101 (2020), p. 235123.