Modeling of First-Order Photobleaching Kinetics Using Krylov Subspace Spectral Methods

Somayyeh Sheikholeslami
*University of Southern Mississippi*

James V. Lambers
*University of Southern Mississippi*, james.lambers@usm.edu

Follow this and additional works at: [https://aquila.usm.edu/fac_pubs](https://aquila.usm.edu/fac_pubs)

**Part of the Mathematics Commons**

**Recommended Citation**


This Article is brought to you for free and open access by The Aquila Digital Community. It has been accepted for inclusion in Faculty Publications by an authorized administrator of The Aquila Digital Community. For more information, please contact Joshua.Cromwell@usm.edu.
Modeling of first-order photobleaching kinetics using Krylov subspace spectral methods

Somayyeh Sheikholeslami, James V. Lambers

\textsuperscript{a}The University of Southern Mississippi, Department of Physics and Astronomy, Hattiesburg, Mississippi, 39406, USA
\textsuperscript{b}The University of Southern Mississippi, Department of Mathematics, Hattiesburg, Mississippi, 39406, USA

Abstract

We solve the first order 2-D reaction-diffusion equations which describe binding-diffusion kinetics using the photobleaching scanning profile of a confocal laser scanning microscope, approximated by a Gaussian laser profile. We show how to solve the first-order photobleaching kinetics partial differential equations (PDEs) using a time-stepping method known as a Krylov subspace spectral (KSS) method. KSS methods are explicit methods for solving time-dependent variable-coefficient partial differential equations. They approximate Fourier coefficients of the solution using Gaussian quadrature rules in the spectral domain. In this paper, we show how a KSS method can be used to obtain not only an approximate numerical solution, but also an approximate analytical solution when using initial conditions that come from pre-bleach steady states and also general initial conditions, to facilitate asymptotic analysis. Analytical and numerical results are presented. It is observed that although KSS methods are explicit, it is possible to use a time step that is far greater than what the CFL condition would indicate.

Keywords: Lanczos algorithm, spectral methods, Gaussian quadrature, FRAP, photobleaching kinetics equation

2010 MSC: 65M70, 65F60

1. Introduction

Fluorescence recovery after photobleaching (FRAP) is a method used to obtain information about the dynamic behavior of the molecules in a cell membrane. A high-intensity laser beam is used to bleach molecules in a region of the cell. The redistribution of the molecules is monitored in both bleached and unbleached regions over time to investigate the movements of molecules within membrane domains. The FRAP method was established by Jacobson et al. in 1976 [11]. The chemical equation of the binding-diffusion process that happens in FRAP is

\[ u + a \xrightarrow{k_{on}} b, \]  

where \( u \) denotes unbound molecules, \( a \) refers to specific binding sites, and \( b \) represents bound complexes (\( ua \)). The rate of the forward binding reaction is called \( k_{on} \), where a molecule binds to
a binding site to form a bound complex, and $k_{off}$ refers to the rate of the reverse unbinding reaction where a molecule is released from its binding site. The first-order reaction-diffusion equations which describe binding-diffusion kinetics are

$$\frac{\partial u}{\partial t} = -k_b I_r(x,y)u + D_1 \Delta u - k_{on}u + k_{off}b \tag{2}$$

$$\frac{\partial b}{\partial t} = -k_b I_r(x,y)b + D_2 \Delta b + k_{on}u - k_{off}b$$

where the initial conditions from the pre-bleach steady state are

$$u(x,y,0) = \frac{k_{off}}{k_{on} + k_{off}} c_i \tag{3}$$

$$b(x,y,0) = \frac{k_{on}}{k_{on} + k_{off}} c_i. \tag{4}$$

$D_1$ and $D_2$ are diffusion coefficients of $u$ and $b$, respectively; $k_{on}$ and $k_{off}$ are the on and off binding-rate constants, $k_b$ is a bleach constant which is the intensity of the bleaching laser, determined from the properties of the fluorophore, and $c_i$ is the initial concentration of the fluorescent molecules inside the bleached zone. Also, $D_1$, $D_2$, $k_{on}$ and $k_{off}$ are positive constants. The photobleaching scanning profile of the confocal of the Gaussian laser can be approximated by \cite{1, 2, 8}

$$I_r(x,y) = \frac{2I_0}{\pi r_n^2} e^{-\frac{2((x - x_c)^2 + (y - y_c)^2)}{r_n^2}}$$

where $r_n$ is the nominal radius of the laser beam and $(x_c, y_c)$ is the center.

First-order photobleaching kinetics which are mathematically modeled in equation (2) were solved numerically by Kang et al. \cite{8, 9}. These equations were also solved numerically using an inversion method (methods of lines, with backward Euler in time and central differencing in space) in \cite{20}. In this paper we apply an explicit time-stepping method known as a Krylov subspace spectral (KSS) method to solve the first-order photobleaching kinetics PDEs. KSS methods developed by Lambers \cite{12} use Gaussian quadrature rules in the spectral domain, as described in \cite{6}, to approximate each Fourier coefficient of the solution. This component-wise approach yields high-order accuracy in time, stability characteristic of implicit methods even though KSS methods are themselves explicit \cite{12, 13}, and superior scalability compared to other time-stepping approaches \cite{4}. We will use a KSS method to solve the first-order photobleaching kinetics equations with initial conditions (3), (4). By applying KSS symbolically to compute each Fourier coefficient, we can also obtain an approximate analytical solution valid for a sufficiently small time step, to facilitate qualitative analysis of the solution.

The outline of the paper is as follows. In Section 2, we describe KSS methods. In Section 3 we use a first-order KSS method to derive formulas for the Fourier coefficients of an approximate solution with general initial data. In Section 4 we derive formulas for the Fourier coefficients of an approximate analytical solution, for sufficiently small time, with initial data obtained from pre-bleach steady states. In Section 5 we explain how this KSS method can be implemented efficiently. Section 6 presents numerical results to validate our approximate analytical solution and demonstrate the accuracy and efficiency of the corresponding numerical method, and Section 7 gives concluding remarks and discussion of future work, including generalizations.
2. Methodology

2.1. Krylov Subspace Spectral Methods

In order to solve equation (2), we apply a Krylov subspace spectral (KSS) method [12] in $[0, 2\pi]^2$ and $t > 0$ with periodic boundary conditions. To describe KSS methods, we scale down to a simpler problem, a single 1-D PDE on $[0, 2\pi]$ rather than a system of two 2-D PDEs:

$$\frac{\partial u}{\partial t} + L(x, D)u = 0, \quad u(x, 0) = u_0(x) \quad u(0, t) = u(2\pi, t)$$

where $D = \partial/\partial x$ and $L(x, D)$ is a differential operator which includes both differentiation operators and coefficients that are functions of $x$. The inner product $\langle \cdot, \cdot \rangle$ is the standard $L^2$ inner product of functions on $[0, 2\pi]$. The Fourier coefficients of the exact solution as inner products are calculated as follows:

$$\hat{u}(\omega, t_{n+1}) = \left\langle \frac{1}{\sqrt{2\pi}} e^{i\omega x}, S(x, D; \Delta t)u(x, t_n) \right\rangle, \quad |\omega| \leq N/2 \quad (5)$$

where $S(x, D; \Delta t) = e^{-L(x, D)\Delta t}$ is the exact solution operator and $N$ is the number of equally-spaced grid points. After spatial discretization, (5) becomes

$$[\hat{u}^{n+1}]_{\omega} = \hat{e}_H^H S_N(\Delta t)u(t_n), \quad S_N = e^{-L_N \Delta t} \quad (6)$$

where $L_N$ is a matrix that represents the spatial discretization of the operator $L(x, D)$. Vector components on a $N$-point grid and uniform grid spacing $h$ are defined by

$$[\hat{e}_{\omega}]_j = \frac{1}{\sqrt{2\pi}} e^{i\omega jh}, \quad [u(t_n)]_j = u(jh, t_n), \quad h = \frac{2\pi}{N} .$$

2.2. Gaussian Quadrature for Riemann-Stieltjes Integrals

The bilinear form in equation (6) that we want to approximate is an example of the generic bilinear form

$$\mathbf{u}^H f(A)\mathbf{v}$$

where $\mathbf{u}$ and $\mathbf{v}$ are $N$-vectors, $f$ is a smooth function and $A = L_N$ is an $N \times N$ symmetric positive definite matrix with positive and real eigenvalues

$$0 < a = \lambda_N \leq \cdots \leq \lambda_2 \leq \lambda_1 = b$$

and also orthonormal eigenvectors $\mathbf{q}_1, \mathbf{q}_2, \ldots, \mathbf{q}_N$ such that

$$A\mathbf{q}_j = \lambda_j \mathbf{q}_j, \quad j = 1, 2, \ldots, N. \quad (9)$$
As shown in [6], the bilinear form (7) can be written as Riemann-Stieltjes integral

\[ u^H f(\lambda)v = u^H f\left(\sum_{j=1}^{N} \lambda_j q_j^H\right)v \]

\[ = \sum_{j=1}^{N} f(\lambda_j)u^H q_j q_j^H v \]

\[ = \int_a^b f(\lambda) d\alpha(\lambda) \]

The measure \( \alpha(\lambda) \) is defined as

\[ \alpha(\lambda) = \begin{cases} 0, & \lambda < a \\ \sum_{j=1}^{N} \alpha_j \beta_j, & \lambda_i \leq \lambda < \lambda_{i-1}, \quad i = 1, \ldots, N - 1 \\ \sum_{j=1}^{N} \alpha_j \beta_j, & b \leq \lambda \end{cases} \]

where \( \alpha_j = u^H q_j \) and \( \beta_j = q_j^H v \). A K-node Gaussian quadrature rule can be written in terms of nodes \( t_j \) and weights \( w_j \) where \( j = 1, 2, \ldots, K \).

\[ u^H f(\lambda)v = \int_a^b f(\lambda) d\alpha(\lambda) = I[f] = \sum_{j=1}^{K} w_j f(t_j) + R[f]. \]

The weights are calculated by

\[ w_j = \int_a^b L_j(\lambda)d\alpha(\lambda), \quad j = 1, 2, \ldots, K, \]

where \( L_j(\lambda) \) is a Lagrange polynomial for the nodes \( t_1, \ldots, t_K \) that can be written

\[ L_j(\lambda) = \prod_{i=1, i \neq j}^{K} \frac{\lambda - t_i}{\lambda_j - t_i}, \quad j = 1, \ldots, K \]

\[ L_j(t_k) = \delta_{jk}. \]

The error can be calculated by

\[ R[f] = \frac{f^{2K}(\eta)}{(2K)!} \int_a^b \left[ \prod_{j=1}^{K} (\lambda - t_j) \right]^2 d\alpha(\lambda), \quad a < \eta < b. \]

2.3. The Case \( u = v \)

In order to construct a Gaussian quadrature rule for the measure \( \alpha(\lambda) \), we define a sequence of polynomials \( q_0(\lambda), q_1(\lambda), \ldots \) that are orthonormal. Orthonormality is defined by the conditions

\[ \int_a^b q_i(\lambda)q_j(\lambda) d\alpha(\lambda) = \delta_{ij} \]
where \( q_j \) has exact degree \( j \) for \( j = 0, 1, 2, \ldots \). We therefore use the inner product we could write

\[
\langle f, g \rangle = \int_a^b f(\lambda)g(\lambda) \, d\alpha(\lambda) = u^H f(A)g(A) v
\]

The polynomials \( q_0, q_1, \ldots \) satisfy a three-term recurrence relationship, which, for \( j = 1, 2, \ldots, K \), can be written

\[
\beta_j q_j(\lambda) = (\lambda - \alpha_j)q_{j-1}(\lambda) - \beta_{j-1} q_{j-2}(\lambda), \quad q_{-1}(\lambda) \equiv 0, \quad q_0(\lambda) \equiv \left( \int_a^b d\alpha(\lambda) \right)^{-1/2},
\]

where for \( j = 1, 2, \ldots, K \) we have

\[
\alpha_j = \langle q_{j-1}, xq_{j-1} \rangle = x^H_j A x_j, \quad j \geq 1
\]

\[
\beta_j = \langle p_j, p_j \rangle^{1/2} = \|r_j\|_2
\]

\[
x_j = q_{j-1}(A) u, \quad j \geq 1
\]

\[
r_j = p_j(A) u = (A - \alpha_j I) q_{j-1}(A) u - \beta_{j-1} q_{j-2}(A) u = (A - \alpha_j I) x_j - \beta_{j-1} x_{j-1}
\]

where \( \alpha_j \) and \( \beta_j \) are recursion coefficients, and \( x_j \) and \( r_j \) are vectors. The resulting orthonormal polynomials and tridiagonal Jacobi matrix \( J_K \), produced by Lanczos iteration, are

\[
q(\lambda) = \begin{bmatrix} q_0(\lambda) \\ q_1(\lambda) \\ \vdots \\ q_{K-1}(\lambda) \end{bmatrix}, \quad J_K = \begin{bmatrix} \alpha_1 & \beta_1 & \beta_2 & \cdots & \beta_{K-2} & \beta_{K-1} \\ \beta_1 & \alpha_2 & \beta_2 & \cdots & \beta_{K-3} & \beta_{K-1} \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ \beta_{K-2} & \beta_{K-3} & \beta_{K-4} & \cdots & \alpha_{K-1} & \beta_{K-1} \\ \beta_{K-1} & \beta_{K-2} & \beta_{K-3} & \cdots & \beta_{K-1} & \alpha_K \end{bmatrix}.
\]

The eigenvalues of \( J_K \) are the nodes for a \( K \)-point Gaussian quadrature rule. The squares of the first elements of the normalized eigenvectors of \( J_K \) are the weights \( w_j = \langle \beta_j q_j(t) \rangle / \|q(t)\|_2^2 \). We then have the Gaussian quadrature approximation

\[
u^H f(A) u = \|u\|_2^2 \|e_1^H f(J_K) e_1\|
\]

which can easily be evaluated in terms of the quadrature nodes and weights.

### 2.4. The Block Case \( u \neq v \)

The block Lanczos iteration \([7]\) produces a block tridiagonal Jacobi matrix \( J_K \) of order \( 2K \), which yields a block Gaussian quadrature rule that can be used for the case \( u \neq v \). We have

\[
J_K = \begin{bmatrix} M_1 & B_1^H \\ B_1 & M_2 & B_2^H \\ \vdots & \vdots & \ddots & \vdots \\ B_{K-2} & M_{K-1} & B_{K-1}^H \\ B_{K-1} & M_K & B_K^H \end{bmatrix}
\]

where, for \( j = 1, 2, \ldots, K \),

\[
R_{j-1} = X_j B_{j-1}, \quad M_1 = X_j^H A X_j, \quad R_j = A X_j - X_j M_j - X_{j-1} B_{j-1}^H
\]

\[5\]
where \( X_j \) is an \( N \times 2 \) matrix, \( X_j^H X_1 = I \), \( B_j \) is \( 2 \times 2 \) upper triangular and \( M_j \) is \( 2 \times 2 \) symmetric [15]. The matrices \( X_j \) and \( B_{j-1} \) are obtained by performing a QR factorization of \( R_{j-1} \). The initial value is \( R_0 = \begin{bmatrix} u & v \end{bmatrix} \). Once \( J_K \) is obtained, a block Gaussian quadrature approximation of (7) is given by

\[
\begin{align*}
\mathbf{u}^H f(A) \mathbf{v} &= e_1 R_0^H E_{11}^H f(J_K) E_{12} B_0 e_2, \\
E_{12} &= \begin{bmatrix} e_1 & e_2 \end{bmatrix}.
\end{align*}
\]

(14)

2.5. Block Arnoldi, \( \mathbf{u} \neq \mathbf{v} \) Case

The spatial differential operator for the system that we are solving is not self-adjoint, and therefore that would be discretized by an unsymmetric matrix. In the case of unsymmetric \( A \), since the orthogonal tridiagonalization does not exist, we could instead obtain a block Hessenberg matrix \( H_K \) in place of the tridiagonal or block tridiagonal Jacobi matrix \( J_K \). The block Hessenberg matrix \( H_K \) of order \( 2K \) for the block case \( \mathbf{u} \neq \mathbf{v} \) can be written

\[
H_K = \begin{bmatrix}
H_{11} & H_{12} & H_{13} & \cdots & H_{1K} \\
H_{21} & H_{22} & H_{23} & \cdots & \cdots \\
\vdots & \ddots & \ddots & \ddots & \vdots \\
0 & 0 & H_{K-1,K-2} & H_{K-1,K-1} & H_{K-1,K} \\
0 & 0 & 0 & H_{K,K-1} & H_{K,K}
\end{bmatrix}
\]

(15)

and for \( j = 1, 2, \ldots, K \) and \( i = 1, 2, \ldots, j \) we have

\[
R_{j-1} = X_i H_{i,j-1}, \quad H_{ij} = X_i^H A X_j, \quad R_j = R_j - X_i H_{ij}
\]

(16)

where \( X_j^H X_1 = I \) [4] and \( R_0 \) is defined as before. As in the block Lanczos case, the eigenvalues and eigenvectors of \( H_K \) are used to construct a quadrature rule for the underlying Riemann-Stieltjes integral, which has the form (14) with \( H_K \) instead of \( J_K \). Such quadrature rules are justified in [4] and the references therein.

The main reason for using block Arnoldi [3, 16] is the loss of orthogonality of the Lanczos vectors in iterations which makes the unsymmetric Lanczos method unstable. In other words, iterations terminates while there is no invariant subspace information for \( A \). Another problem with the unsymmetric Lanczos method is lack of convergence of eigenvalues and also serious breakdown [5].

For each time step and each frequency \( \omega \), the block KSS method proceeds by defining

\[
R_0 = \begin{bmatrix} u & v \end{bmatrix} = \frac{1}{\sqrt{2\pi}} e^{i\omega x_N} u(x_N, t_0)
\]

as the initial block for the block Arnoldi algorithm [5] described above, where \( x_N \) is a vector of equally spaced grid points. We compute the QR factorization \( R_{j-1} = X_i H_{i,j-1} \). Then, block Arnoldi [3, 16] is applied to produce the Hessenberg matrix \( H_K \), which in turn yields the nodes and weights for the Gaussian quadrature rule needed to approximate each Fourier coefficient of the solution at time \( t_{n+1} \). The details of these steps are discussed in the next two sections.

The temporal order of accuracy which has been reported for KSS methods applied to the heat equation [12], the wave equation [13], the Schrödinger equation [14], and Maxwell’s equations [15] are \( O(\Delta t^{2(2K-1)^3/d}) \), where \( d \) is the highest order of a time derivative in the PDE. For the case \( K = 1 \), unconditional stability was proved for the parabolic [12] and hyperbolic [13] PDEs in 1-D, even though KSS methods are explicit. We will investigate the stability and convergence of the single-block node KSS method, as applied to equation (2), in future work.
3. Application of a First-Order KSS Method

In this section, we present the details of applying a first-order ($K = 1$) KSS method to (2). We seek a solution that is a linear combination of chosen basis functions, in which each coefficient in the linear combination is an expression of the form (7) that can be approximated using the approach outlined in Section 2. We begin with the construction of appropriate basis functions.

3.1. Construction of Basis Functions

For convenience, we use the spatial domain $E = [0, 2\pi]^2$, and impose periodic boundary conditions, as the initial data consists of constant functions and the evolution of the solution takes place in the interior of the domain. Homogeneous Dirichlet or Neumann boundary conditions can be handled in a similar manner [4]; we will discuss this in detail in Section 6.3. With this domain and boundary conditions, we use as our basis functions the eigenfunctions of a constant-coefficient problem obtained by averaging the variable coefficient $k_b I_n$, as described in [15].

Let $\omega = (\omega_1, \omega_2) \in \mathbb{Z}^2$. We denote points in $E$ by $x = (x, y)$. We define

$$L = \begin{bmatrix} -k_b I_n(x) - k_{on} + D_1 \Delta & k_{off} \\ k_{off} & -k_b I_n(x) - k_{off} + D_2 \Delta \end{bmatrix}, \quad \mathbf{v}(t) = \begin{bmatrix} u(x, t) \\ b(x, t) \end{bmatrix}, \quad (17)$$

then (2) takes the form $\mathbf{v}_t = L\mathbf{v}$. We now seek to solve a constant-coefficient approximation of this system of PDEs. To that end, the average of $I_n(x)$ over a rectangular domain $E$ is given by

$$\bar{I}_n = \frac{1}{A(E)} \int_E I_n(x, y) \, dx \, dy,$$

We then compute the $2 \times 2$ matrix

$$\bar{L}(\omega) = \begin{bmatrix} \langle e^{i\omega x}, \bar{L}_{11} e^{i\omega x} \rangle & \langle e^{i\omega x}, \bar{L}_{12} e^{i\omega x} \rangle \\ \langle e^{i\omega x}, \bar{L}_{21} e^{i\omega x} \rangle & \langle e^{i\omega x}, \bar{L}_{22} e^{i\omega x} \rangle \end{bmatrix},$$

where $\langle \cdot, \cdot \rangle$ is the standard $L^2$ inner product on $E$ and, for $i, j = 1, 2$, $\bar{L}_{ij}$ is obtained from the entry $L_{ij}$ of $L$ by replacing $I_n$ by $\bar{I}_n$. Then, the eigenvalues of $\bar{L}(\omega)$ are

$$\lambda_1 = -\frac{k_b I_0}{A(E)} - \frac{1}{2} ||\omega||^2 (D_1 + D_2) - \frac{1}{2} (k_{on} + k_{off}) + $$

$$\frac{1}{2} \left[ ||\omega||^2 (D_1 - D_2)^2 + ||\omega||^2 (2(D_1 + D_2)(k_{on} + k_{off}) - 

$$4(D_1 k_{off} + D_2 k_{on})) + (k_{on} + k_{off})^2 \right]^{1/2}$$

$$\lambda_2 = -\frac{k_b I_0}{A(E)} - \frac{1}{2} ||\omega||^2 (D_1 + D_2) - \frac{1}{2} (k_{on} + k_{off}) - 

\frac{1}{2} \left[ ||\omega||^2 (D_1 - D_2)^2 + ||\omega||^2 (2(D_1 + D_2)(k_{on} + k_{off}) - 

$$4(D_1 k_{off} + D_2 k_{on})) + (k_{on} + k_{off})^2 \right]^{1/2}.$$ 

If we write

$$\begin{bmatrix} u_{11} \\ u_{21} \end{bmatrix} = \begin{bmatrix} k_{off} \\ k_{off} \end{bmatrix} \left( \begin{array}{c} \frac{k_b I_0}{A(E)} + D_1 ||\omega||^2 + k_{on} + \lambda_1 \\ 7 \end{array} \right) \quad (18)$$
then one eigenvector of $\bar{L}(\omega)$ is
\[
\begin{bmatrix}
u_{11} \\ u_{21}
\end{bmatrix} = \begin{bmatrix} k_{\text{off}} \\ c_1
\end{bmatrix}
\]
where
\[
c_1 = \frac{1}{2} ||\omega||^2(D_1 - D_2) + \frac{1}{2}(k_{\text{on}} - k_{\text{off}}) + \\
\frac{1}{2} \left[ ||\omega||^4(D_1 - D_2)^2 + ||\omega||^2(2(D_1 + D_2)(k_{\text{on}} + k_{\text{off}}) - \\
4(D_1k_{\text{off}} + D_2k_{\text{on}})) + (k_{\text{on}} + k_{\text{off}})^2 \right]^{1/2}.
\]
If the eigenvalues $\lambda_1$ and $\lambda_2$ are distinct, then the eigenvector corresponding to $\lambda_2$ is given by
\[
\begin{bmatrix}
u_{12} \\ u_{22}
\end{bmatrix} = \begin{bmatrix} k_{\text{off}} \\ c_2
\end{bmatrix}
\]
where
\[
c_2 = \frac{1}{2} ||\omega||^2(D_1 - D_2) + \frac{1}{2}(k_{\text{on}} - k_{\text{off}}) - \\
\frac{1}{2} \left[ ||\omega||^4(D_1 - D_2)^2 + ||\omega||^2(2(D_1 + D_2)(k_{\text{on}} + k_{\text{off}}) - \\
4(D_1k_{\text{off}} + D_2k_{\text{on}})) + (k_{\text{on}} + k_{\text{off}})^2 \right]^{1/2}.
\]
Let $U$ be the $2 \times 2$ matrix with entries $u_{ij}$, $j = 1, 2$. By computing $V = U^{-T}$, we obtain the left eigenvectors of $\bar{L}(\omega)$:
\[
U^{-1}(\omega) = V^T(\omega) = \frac{1}{k_{\text{off}}(c_2 - c_1)} \begin{bmatrix} c_2 & -c_1 \\ -k_{\text{off}} & k_{\text{off}}
\end{bmatrix}
\]
\[
= \frac{-1}{(D_1 - D_2)(\omega_1^2 + \omega_2^2) + (k_{\text{on}} - k_{\text{off}})} \begin{bmatrix} \omega_2 & -\omega_1 \\ k_{\text{off}} & -k_{\text{off}}
\end{bmatrix}
\]
If we write
\[
U(\omega) = \begin{bmatrix} u_1(\omega) & u_2(\omega)
\end{bmatrix}
\]
and similar for $V(\omega)$, then the right and left eigenfunctions, respectively, of the frozen-coefficient operator $\bar{L}$ are, for $j = 1, 2$,
\[
u_j(\omega) = u_j(\omega) \otimes e^{i(\omega \cdot x)} = \begin{bmatrix} u_{1j}e^{i(\omega \cdot x)} \\ u_{2j}e^{i(\omega \cdot x)}
\end{bmatrix}
\]
\[
v_j(\omega) = v_j(\omega) \otimes e^{i(\omega \cdot x)} = \begin{bmatrix} v_{1j}e^{i(\omega \cdot x)} \\ v_{2j}e^{i(\omega \cdot x)}
\end{bmatrix}
\]
If the eigenvalues $\lambda_1$ and $\lambda_2$ are equal, we use Schur vectors instead of eigenvectors, which entails setting
\[
\begin{bmatrix}
u_{12} \\ u_{22}
\end{bmatrix} = \begin{bmatrix} -c_1 \\ k_{\text{off}}
\end{bmatrix}
\]
and computing \( V(\omega) = U(\omega)/(k_0^2 + \omega^2) \).

Once we have obtained our basis functions for each \( \omega \in \Omega \subset \mathbb{R}^2 \), where \( \Omega \) is a chosen subset of frequency space determined by spatial discretization, we then seek a solution of (2) of the form

\[
\begin{bmatrix}
  u(x, t_{n+1}) \\
  b(x, t_{n+1})
\end{bmatrix} = \sum_{\omega \in \Omega} \sum_{j=1}^{2} \mathbf{u}_{j\omega}(x) \begin{bmatrix}
  v_{j\omega, e^x} \\
  b(x, t_n)
\end{bmatrix},
\]

where \( \Delta t = t_{n+1} - t_n \) is a chosen time step. We can then approximate each inner product in the above linear combination by treating it as a Riemann-Stieltjes integral, as described in Section 2.

### 3.2. Block Arnoldi, \( \omega \neq 0 \)

Now that our basis functions have been chosen, we will approximate each coefficient of the form

\[
\begin{bmatrix}
  v_{j\omega}, u(x, t_{n+1}) \\
  b(x, t_{n+1})
\end{bmatrix} = \begin{bmatrix}
  v_{j\omega}, e^{\Delta t \omega} \\
  b(x, t_n)
\end{bmatrix}, \quad j = 1, 2, \quad (19)
\]

using a two-node block Gaussian quadrature rule. We begin with spatial discretization, using a uniform grid with spacing \( \Delta x = \Delta y = 2\pi/N \), where for convenience we assume \( N \) is even. We denote by \( x_N \) a \( N^2 \times 2 \) matrix, the columns of which contain the \( x- \) and \( y-\) coordinates, respectively, of the grid points \( (x_i, y_j), i, j = 0, 1, \ldots, N - 1 \), with \( x_i = i\Delta x \) and \( y_j = j\Delta y \). As before, we also denote by \( \omega = (\omega_1, \omega_2) \) a pair of wave numbers, where \( -N/2 + 1 \leq \omega_i \leq N/2 \) for \( i = 1, 2 \). It follows that spatial error is introduced due to the truncation of the Fourier series of \( u(x, t) \) and \( b(x, t) \).

To approximate each coefficient (19), we perform a single iteration of block Arnoldi, with \( N^2 \times 2 \) initial blocks

\[
\begin{align*}
R_0 &= \begin{bmatrix}
  v_{11}(\omega) e^{i\omega x_0} & u(x_N, t_n) \\
  v_{21}(\omega) e^{i\omega x_0} & b(x_N, t_n)
\end{bmatrix} \\
\tilde{R}_0 &= \begin{bmatrix}
  v_{12}(\omega) e^{i\omega x_0} & u(x_N, t_n) \\
  v_{22}(\omega) e^{i\omega x_0} & b(x_N, t_n)
\end{bmatrix}.
\end{align*}
\]

Before proceeding, we introduce the following notation for conciseness. First, we suppress the explicit dependence on \( x_N \) and \( t_n \) in the current solution:

\[
u = u(x_N, t_n), \quad b = b(x_N, t_n).
\]

Next, we use a similar shorthand for the discrete Fourier transforms, which implicitly depend on \( \omega \):

\[
\hat{\nu} = (e^{i\omega x_0})^H u, \quad \hat{b} = (e^{i\omega x_0})^H b.
\]

Throughout, we use \( \| \cdot \| \) to refer to vector \( \ell^2 \)-norms.

Then, as the first step in block Arnoldi, we perform a QR factorization of \( R_0 \) and \( \tilde{R}_0 \), which yields

\[
R_0 = X_0 B_0, \quad \tilde{R}_0 = \tilde{X}_0 \tilde{B}_0.
\]

By defining

\[
\begin{align*}
f &= uN^2\|v_1(\omega)\|^2 - (v_{11}^2(\omega)\hat{\nu} + v_{21}(\omega)\hat{h})e^{i\omega x_0} \\
g &= bN^2\|v_1(\omega)\|^2 - (v_{21}^2(\omega)\hat{\nu} + v_{22}(\omega)\hat{h})e^{i\omega x_0} \\
h &= uN^2\|v_2(\omega)\|^2 - (v_{12}^2(\omega)\hat{\nu} + v_{22}(\omega)\hat{h})e^{i\omega x_0} \\
k &= bN^2\|v_2(\omega)\|^2 - (v_{22}^2(\omega)\hat{\nu} + v_{22}(\omega)\hat{h})e^{i\omega x_0}
\end{align*}
\]
we obtain
\[
X_1 = \begin{bmatrix} x_{11} & x_{12} \end{bmatrix} = \begin{bmatrix} v_{11}(\omega) e^{i\omega x} \\ v_{21}(\omega) e^{i\omega x} \end{bmatrix} \frac{f}{N[\|v_1(\omega)\|^2]} + \begin{bmatrix} g \\ h \end{bmatrix} \frac{\sqrt{[f]^2 + [g]^2}}{\sqrt{[f]^2 + [g]^2}}
\]
\[
\tilde{X}_1 = \begin{bmatrix} \tilde{x}_{11} & \tilde{x}_{12} \end{bmatrix} = \begin{bmatrix} v_{12}(\omega) e^{i\omega x} \\ v_{22}(\omega) e^{i\omega x} \end{bmatrix} \frac{h}{N[\|v_2(\omega)\|^2]} + \begin{bmatrix} k \end{bmatrix} \frac{\sqrt{[h]^2}}{\sqrt{[h]^2}}.
\]
(20)

It should be noted that if the columns of $R_0$ or $\tilde{R}_0$ are not linearly independent, then we can simply use the approach described in Section 2.3 to approximate our bilinear form (19), which is in this case effectively a quadratic form.

Then, $M_1$ and $\bar{M}_1$, which correspond to $H_{11}$ from the block Arnoldi algorithm, can be calculated by
\[
M_1 = X_1^H L X_1 = \begin{bmatrix} M_{11} & M_{12} \\ M_{21} & M_{22} \end{bmatrix}
\]
(21)
\[
\bar{M}_1 = \bar{X}_1^H L \bar{X}_1 = \begin{bmatrix} \bar{M}_{11} & \bar{M}_{12} \\ \bar{M}_{21} & \bar{M}_{22} \end{bmatrix}.
\]
(22)

Formulas for the entries of $M_1$ and $\bar{M}_1$ are given in Section Appendix A.1 of the Appendix.

After computing the eigenvalues $\lambda_{1,\omega}$, $\lambda_{2,\omega}$ of $M_1$, the coefficient of $u_{1,\omega}$ in the solution at time $t$ can be approximated by
\[
[B_0^H e^{M_{1,\omega} t} B_0]_{12} = \left\langle v_1(\omega) \otimes e^{i\omega x}, \rho_\omega(L_N) \begin{bmatrix} u \\ b \end{bmatrix} \right\rangle
\]
where $\rho_\omega$ is a polynomial of degree 1 which interpolates $e^{i\omega t}$ at $\lambda_{1,\omega}$ and $\lambda_{2,\omega}$ and is given by
\[
\rho_\omega(L_N) = \frac{e^{i\lambda_{1,\omega} t} - L_N - \lambda_{1,\omega} I}{\lambda_{1,\omega} - \lambda_{2,\omega}} + \frac{e^{i\lambda_{2,\omega} t} - L_N - \lambda_{2,\omega} I}{\lambda_{2,\omega} - \lambda_{1,\omega}}
\]
if $\lambda_{1,\omega} \neq \lambda_{2,\omega}$, based on Lagrange interpolation, or
\[
\rho_\omega(L_N) = e^{i\lambda_{1,\omega} t} + \Delta t e^{i\lambda_{1,\omega} t} (L_N - \lambda_{1,\omega} I)
\]
if $\lambda_{1,\omega} = \lambda_{2,\omega}$, based on Hermite interpolation. The coefficient of $u_{2,\omega}$ can be approximated in a similar manner, using the eigenvalues $\bar{\lambda}_{1,\omega}, \bar{\lambda}_{2,\omega}$ of $\bar{M}_1$.

The value of $\Delta t$ can be chosen sufficiently small to ensure desired accuracy, and then this process can be repeated in subsequent time steps. By using the integrand $g(\lambda) = \lambda e^{i\lambda t}$ in place of $f(\lambda) = e^{i\lambda t}$, one can easily use the above $M_1$ and $\bar{M}_1$ to compute an approximate time derivative, which can then be used to obtain a residual $v_i - L v_i$. This residual can serve as an estimate of local truncation error, for the purpose of adaptive time-stepping.
3.3. Block Arnoldi, $\omega_1 = \omega_2 = 0$

We now use block Arnoldi for the case $\omega_1 = \omega_2 = 0$, for which we use the initial block

$$ R_0 = \begin{bmatrix} -u_{22} & u(x_N, t_n) \\ k_{\text{off}}(k_{\text{on}} - k_{\text{off}}) & b(x_N, t_n) \end{bmatrix}. $$

Then, after computing the QR factorization $R_0 = X_1 B_0$, we have

$$ X_1 = \begin{bmatrix} \frac{1}{\sqrt{2N}} u - \bar{b} + \bar{u} \\ \frac{1}{\sqrt{2N}} b - \bar{b} + \bar{u} \end{bmatrix}, $$

where we use $\bar{f}$ to denote the average of the components of a grid function $f$. Then, we compute $M_1 = X_1^H L X_1$. Formulas for the entries of $M_1$ are given in Appendix A.2.

Similarly, for the coefficient of $u_{2(0,0)}$, our initial block is

$$ R_0 = \begin{bmatrix} k_{\text{on}}(k_{\text{on}} - k_{\text{off}}) & u(x_N, t_n) \\ k_{\text{off}} & b(x_N, t_n) \end{bmatrix}. $$

Computing the QR factorization $R_0 = \tilde{X}_1 \tilde{B}_0$ yields

$$ \tilde{X}_1 = \begin{bmatrix} \frac{k_{\text{on}}}{\sqrt{d_{21}}} \left( u(k_{\text{on}}^2 + k_{\text{off}}^2) - k_{\text{on}}k_{\text{off}}\bar{b} \right) \\ \frac{-k_{\text{off}}}{\sqrt{d_{21}}} \left( b(k_{\text{on}}^2 + k_{\text{off}}^2) - k_{\text{on}}k_{\text{off}}\bar{u} \right) \end{bmatrix}, $$

$$ d_{21} = ||u||^2 + ||b||^2 - N^2 \bar{b}^2 \bar{u}^2 - \frac{k_{\text{on}}^2 k_{\text{off}}^2}{k_{\text{on}}^2 + k_{\text{off}}^2} - \frac{k_{\text{on}}^2 k_{\text{off}}^2}{k_{\text{on}}^2 + k_{\text{off}}^2} + \frac{2N^2 \bar{b} \bar{u} k_{\text{on}} k_{\text{off}}}{k_{\text{on}}^2 + k_{\text{off}}^2}. $$

Then, we compute $\tilde{M}_1 = \tilde{X}_1^H L \tilde{X}_1$. Formulas for the entries of $\tilde{M}_1$ are given in Appendix A.2.

4. Approximate Analytical Solution

In this section, we specialize to initial data (3), (4) from pre-bleach steady states to obtain an approximate analytical solution for sufficiently small $t$. The terms in the entries of $V(\omega)$ that are of lower order in $||\omega||$ are neglected.
4.1. The $\omega \neq 0$ Case

To facilitate analysis of high-frequency components, here we neglect lower-order terms in $||\omega||_2$. For the component of the solution in the directions of $u_{1,\omega}$ and $u_{2,\omega}$ with $||\omega|| > 0$, the initial blocks are

\[
R_0 = \begin{bmatrix}
-1 & u_{12} e^{i\omega x_1} & \frac{k_{\text{off}}}{k_{\text{off}} + k_{\text{on}}} \\
(D_1 - D_2)||\omega||^2 + (k_{\text{on}} - k_{\text{off}}) & -e^{i\omega x_1} & \frac{k_{\text{off}} C_{1}}{k_{\text{off}} + k_{\text{on}}} \\
\end{bmatrix}, \quad \tilde{R}_0 = \begin{bmatrix}
-1 & u_{12} e^{i\omega x_1} & \frac{k_{\text{off}}}{k_{\text{off}} + k_{\text{on}}} \\
(D_1 - D_2)||\omega||^2 + (k_{\text{on}} - k_{\text{off}}) & -e^{i\omega x_1} & \frac{k_{\text{off}} C_{1}}{k_{\text{off}} + k_{\text{on}}} \\
\end{bmatrix}
\]

Orthogonalization of these initial blocks yields

\[
X_1 = \frac{1}{N \sqrt{u_{22}^2 + k_{\text{off}}^2}} \begin{bmatrix}
-u_{22} e^{i\omega x_1} & k_{\text{off}} \\
k_{\text{off}} e^{i\omega x_1} & u_{22}
\end{bmatrix}, \quad \tilde{X}_1 = \frac{1}{N \sqrt{u_{21}^2 + k_{\text{off}}^2}} \begin{bmatrix}
-u_{21} e^{i\omega x_1} & k_{\text{off}} \\
k_{\text{off}} e^{i\omega x_1} & u_{21}
\end{bmatrix}.
\]

We then compute $M_1 = X_1^H L X_1$ and $\tilde{M}_1 = \tilde{X}_1^H L \tilde{X}_1$. Formulas for the entries of $M_1$ and $\tilde{M}_1$ are given in Appendix A.3.

As before, for $j = 1, 2$, let $\lambda_{j,\omega}$ and $\tilde{\lambda}_{j,\omega}$ denote the eigenvalues of $M_1$ and $\tilde{M}_1$, respectively. Formulas for these eigenvalues are also given in Appendix A.3. Then, the component of the solution in the direction of $u_{1,\omega}$ is

\[
[B_{12}^H e^{M_1 t} R_0]_{12} = \begin{bmatrix}
\vec{v}_1(\omega) \otimes e^{i\omega x} & \rho_{\omega}(L) \begin{bmatrix} u(x, 0) \\ b(x, 0) \end{bmatrix}
\end{bmatrix}
\]

where $\rho_{\omega}$ is a polynomial of degree 1 which interpolates $e^{i\omega t}$ at $\lambda_{1,\omega}$ and $\lambda_{2,\omega}$ given by (23) or (24), depending on whether $\lambda_{1,\omega}$ and $\lambda_{2,\omega}$ are distinct. The component in the direction of $u_{2,\omega}$ can be computed similarly, by interpolating at $\lambda_{1,\omega}$ and $\lambda_{2,\omega}$. We conclude that the approximate analytical solution is

\[
\begin{bmatrix}
\begin{bmatrix} u(x, t) \\ b(x, t) \end{bmatrix} = \sum_{\omega \in \Omega} u_{1,\omega}(x) \begin{bmatrix} \vec{v}_1(\omega) \otimes e^{i\omega x} \\ \rho_{\omega}(L) \end{bmatrix} \begin{bmatrix} u(x, 0) \\ b(x, 0) \end{bmatrix} \\
\sum_{\omega \in \Omega} u_{2,\omega}(x) \begin{bmatrix} \vec{v}_2(\omega) \otimes e^{i\omega x} \\ \tilde{\rho}_{\omega}(L) \end{bmatrix} \begin{bmatrix} u(x, 0) \\ b(x, 0) \end{bmatrix} + \\
\end{bmatrix} = \begin{bmatrix}
\frac{-1}{(D_1 - D_2)||\omega||^2 + (k_{\text{on}} - k_{\text{off}})} & \begin{bmatrix} u_{11}(\omega) e^{i\omega x} \\ u_{21}(\omega) e^{i\omega x} \end{bmatrix} \\
\frac{-1}{(D_1 - D_2)||\omega||^2 + (k_{\text{on}} - k_{\text{off}})} & \begin{bmatrix} u_{12}(\omega) e^{i\omega x} \\ u_{22}(\omega) e^{i\omega x} \end{bmatrix}
\end{bmatrix} \times \\
\begin{bmatrix}
\frac{k_{\text{off}} C_{1}}{k_{\text{off}} + k_{\text{on}}} \\
\frac{k_{\text{off}} C_{1}}{k_{\text{off}} + k_{\text{on}}}
\end{bmatrix} \begin{bmatrix}
\tilde{\rho}_{\omega}(L) \\
\tilde{\rho}_{\omega}(L)
\end{bmatrix}
\]

\[
\frac{1}{k_{\text{off}}^2} e^{-i\omega x} e^{i\omega x} \begin{bmatrix}
\tilde{\rho}_{\omega}(L) \\
\tilde{\rho}_{\omega}(L)
\end{bmatrix} \frac{k_{\text{off}} C_{1}}{k_{\text{off}} + k_{\text{on}}} \begin{bmatrix}
\tilde{\rho}_{\omega}(L) \\
\tilde{\rho}_{\omega}(L)
\end{bmatrix}.
\]

(25)
Here, we used exact basis functions which are valid for all frequencies, while in previous work [4] basis functions were approximated for the constant coefficient problem which were valid only at high frequencies.

4.2. The $\omega_1 = \omega_2 = 0$ Case

To complete the approximate analytical solution (25), we consider the special case $\omega_1 = \omega_2 = 0$. Our initial block for the component in the direction of $u_1(0,0)$ is

$$R_0 = \begin{bmatrix} \frac{u_{22}}{k_{\text{off}}(k_{\text{on}} - k_{\text{off}})} & \frac{k_{\text{off}} c_i}{k_{\text{on}} + k_{\text{off}}} \\ \frac{k_{\text{off}} c_i}{k_{\text{on}} + k_{\text{off}}} & \frac{1}{k_{\text{off}}} \end{bmatrix}.$$ 

Substituting $u_{22} = -k_{\text{off}}$ in $x_{11}$ and $x_{12}$, $X_1$ becomes

$$X_1 = \begin{bmatrix} \frac{1}{\sqrt{2N}} & -\frac{1}{\sqrt{2N}} \\ \frac{1}{\sqrt{2N}} & \frac{1}{\sqrt{2N}} \end{bmatrix}$$

which yields

$$M_1 = X_1^H L X_1 = \begin{bmatrix} -k_{\text{off}} T_{r_{\text{on}}} & 0 \\ k_{\text{on}} - k_{\text{off}} & -k_{\text{off}} T_{r_{\text{on}}} - (k_{\text{on}} + k_{\text{off}}) \end{bmatrix}.$$ 

The eigenvalues of $M_1$ are

$$\lambda_{1,(0,0)} = -k_{\text{off}} T_{r_{\text{on}}}$$

$$\lambda_{2,(0,0)} = -k_{\text{off}} T_{r_{\text{on}}} - (k_{\text{on}} + k_{\text{off}}).$$

For the component of the solution in the direction of $u_{2, (0,0)}$, we use the initial block

$$\tilde{R}_0 = \begin{bmatrix} u_{21} \\ k_{\text{off}}(k_{\text{on}} - k_{\text{off}}) \end{bmatrix} \begin{bmatrix} k_{\text{off}} c_i \\ k_{\text{on}} - k_{\text{off}} \end{bmatrix}$$

where the QR factorization variables obtained by substituting $u_{21} = k_{\text{on}}$ are

$$\tilde{X}_1 = \begin{bmatrix} \tilde{x}_{11} & \tilde{x}_{12} \end{bmatrix} = \begin{bmatrix} \frac{k_{\text{on}}}{N \sqrt{(k_{\text{on}}^2 + k_{\text{off}}^2)}} & \frac{k_{\text{off}}}{N \sqrt{(k_{\text{on}}^2 + k_{\text{off}}^2)}} \\ \frac{k_{\text{off}}}{N \sqrt{(k_{\text{on}}^2 + k_{\text{off}}^2)}} & \frac{k_{\text{on}}}{N \sqrt{(k_{\text{on}}^2 + k_{\text{off}}^2)}} \end{bmatrix}.$$ 

Then

$$M_1 = \tilde{X}_1^H L \tilde{X}_1 = \begin{bmatrix} -k_{\text{off}} T_{r_{\text{on}}} - (k_{\text{on}} + k_{\text{off}}) & 0 \\ k_{\text{on}} - k_{\text{off}} & -k_{\text{off}} T_{r_{\text{on}}} \end{bmatrix}.$$
It follows that the eigenvalues of $\hat{M}_1$ are
\[
\hat{\lambda}_{1,0,0} = -k_0 \bar{T}_{\lambda} -(k_{\text{nn}} + k_{\text{df}})
\]
\[
\hat{\lambda}_{2,0,0} = -k_0 \bar{T}_{\lambda}.
\]

We then use the pairs of interpolation points $(A_{1,0,0}, A_{2,0,0})$ and $(\hat{\lambda}_{1,0,0}, \hat{\lambda}_{2,0,0})$, as in the $\omega \neq 0$ case, to compute the coefficients of the solution (25) in the directions of $u_{1,0,0}$ and $u_{2,0,0}$, respectively.

5. Implementation Details

In this section we show how the KSS method for modeling first-order photobleaching kinetics can be implemented efficiently through vectorized polynomial interpolation. This approach can be used to produce a numerical solution through time-stepping, using the more general formulas for the block Gaussian quadrature nodes given in Section 4, or a spectral accuracy in space. Regardless, the main error is the first-order temporal error.

To carry out a time step from time $t_n$ to time $t_{n+1}$, we first compute, for each $\omega = (\omega_1, \omega_2)$ with $\omega_i$ in the indicated range, the block Gaussian quadrature nodes $\lambda_{1,\omega}, \lambda_{2,\omega}, \tilde{\lambda}_{1,\omega}, \tilde{\lambda}_{2,\omega}$ from Section 3. Next, we compute the first-degree polynomial $\rho_{\omega}(\lambda)$ that interpolates $e^{i\lambda t}$ at $\lambda_{1,\omega}$ and $\lambda_{2,\omega}$. The coefficients of $\rho_{\omega}(\lambda)$ in power form are given by
\[
c_{1,\omega} = \begin{cases} 
\frac{f_{2,\omega} - f_{1,\omega}}{\Delta f_{1,\omega}} & \lambda_{1,\omega} \neq \lambda_{2,\omega}, \\
\frac{\lambda_{2,\omega} - \lambda_{1,\omega}}{\Delta f_{1,\omega}} & \lambda_{1,\omega} = \lambda_{2,\omega},
\end{cases}
\]
\[
c_{0,\omega} = f_{1,\omega} - c_{1,\omega} \lambda_{1,\omega},
\]
where
\[
f_{k,\omega} = e^{ik\omega t}, \quad k = 1, 2.
\]
The polynomial $\tilde{\rho}_{\omega}(\lambda)$ that interpolates $e^{i\lambda t}$ at $\lambda_1$ and $\lambda_2$ is computed in a similar manner. We denote its coefficients by $\tilde{c}_{0,\omega}$ and $\tilde{c}_{1,\omega}$.

Next, we compute
\[
\rho_{\omega}(L_N) \begin{bmatrix} u \\ b \end{bmatrix}_n = c_{0,\omega} \begin{bmatrix} u \\ b \end{bmatrix}_n + c_{1,\omega} \begin{bmatrix} p \\ q \end{bmatrix}_n,
\]
\[
\tilde{\rho}_{\omega}(L_N) \begin{bmatrix} u \\ b \end{bmatrix}_n = \tilde{c}_{0,\omega} \begin{bmatrix} u \\ b \end{bmatrix}_n + \tilde{c}_{1,\omega} \begin{bmatrix} p \\ q \end{bmatrix}_n,
\]
where
\[
\begin{bmatrix} u \\ b \end{bmatrix}_n = \begin{bmatrix} u(x, t_n) \\ b(x, t_n) \end{bmatrix}, \quad \begin{bmatrix} p \\ q \end{bmatrix}_n = L_N \begin{bmatrix} u \\ b \end{bmatrix}_n.
\]
The application of the operator $L$ through the matrix $L_N$ can be performed using a finite difference discretization of the Laplacian, using the standard five-point stencil on our uniform grid, or using a 2-D FFT. The former approach is more efficient, but the latter yields spectral accuracy in space, rather than second-order accuracy. Regardless, the main error is the first-order temporal error.
Finally, the Fourier coefficients of the solution at time \( t_{n+1} \) are computed as follows:

\[
\begin{bmatrix}
\hat{u}(\omega, t_{n+1}) \\
\hat{b}(\omega, t_{n+1})
\end{bmatrix}
= \begin{bmatrix}
u_{11}(\omega) \\
u_{21}(\omega)
\end{bmatrix}
\begin{bmatrix}u \\ b\end{bmatrix} + \begin{bmatrix}u_{12}(\omega) \\
u_{22}(\omega)
\end{bmatrix}
\begin{bmatrix}v_{1,\omega} \cdot \hat{\rho}_{\omega}(L_N) \\
v_{2,\omega} \cdot \hat{\rho}_{\omega}(L_N)
\end{bmatrix}
\begin{bmatrix}u \\ b\end{bmatrix}
\]

To compute \( \hat{u} \) and \( \hat{b} \), matrices of 2-D Fourier coefficients are multiplied component-wise. An inverse FFT yields the solution at time \( t_{n+1} \). If we are only taking a single time step, the process described in this section is still used, but with the formulas for the nodes given in Section 4.

The resulting algorithm requires \( O(N^2 \log N) \) floating-point operations per time step, where \( N \) is the number of grid points per dimension. While other methods that use a finite difference discretization require only \( O(N^2) \) floating-point operations per time step, it will be seen in Section 6.2 that an accurate solution can still be obtained via KSS with much greater efficiency.

### 6. Numerical Results

We now use numerical experiments to validate the formulas of the previous sections and examine the performance, in terms of accuracy, efficiency, and scalability, of a first-order KSS method applied to (2).

#### 6.1. Validation of Time-Stepping Scheme

First, we use the formulas of Section 3 to implement a time-stepping method, as described in Section 5. We present the relative error versus number of grid points per dimension (\( N \)) and number of time steps (\( n\text{steps} \)). Errors are computed by comparing the solution at the final time (denoted by \( t_f \)) to that obtained by computing \( e^{t_f \omega} \) times the initial data, using the MATLAB function \texttt{expm}. In all cases, we use the parameter values \( k_0 = 1 \) and \( c_i = 1 \). The center \((x_c, y_c)\) of the laser profile is set to be \((\pi, \pi)\).

<table>
<thead>
<tr>
<th>( N )</th>
<th>Execution time (s)</th>
<th>Relative error</th>
</tr>
</thead>
<tbody>
<tr>
<td>8</td>
<td>4.4901</td>
<td>0.0012</td>
</tr>
<tr>
<td>16</td>
<td>7.9654</td>
<td>1.8657e-05</td>
</tr>
<tr>
<td>32</td>
<td>19.7626</td>
<td>3.6038e-06</td>
</tr>
<tr>
<td>64</td>
<td>75.9164</td>
<td>6.9968e-08</td>
</tr>
</tbody>
</table>

For our first test case, reaction-dominant parameters that are defined in Chapter 1 are set to be \( k_{\text{on}} = 10^{-0.5} \, \text{s}^{-1}, k_{\text{off}} = 10^{-1} \, \text{s}^{-1}, D_1 = 30 \, \mu\text{m}^2/\text{s}, D_2 = 10^{-4} \, \mu\text{m}^2/\text{s} \) and \( \omega_0 = 0.5 \, \mu\text{m} \) for Figure 1. These values are taken from [8, 10, 18]. Figure 1a shows relative error versus grid points (\( N \)) for \( n\text{steps} = 10,000 \) (number of time steps) and final time \( t_f = 1 \). It shows a rapidly decreasing trend for relative error with increasing \( N \). This is due to each solution being compared to an approximate solution computing using the matrix exponential on a finer grid. Figure 1b shows relative error versus time steps for \( N = 64 \) and final time \( t_f = 1 \), corresponding to a time step \( \Delta t = 1/n\text{steps} \). It
Figure 1: a: Relative error versus grid points (N). b: Relative error versus time steps. Parameters are $k_{on} = 10^{-0.5} \text{s}^{-1}$, $k_{off} = 10^{-1} \text{s}^{-1}$, $D_1 = 30 \mu\text{m}^2/\text{s}$, $D_2 = 10^{-4} \mu\text{m}^2/\text{s}$ and $\omega_m = 0.5 \mu\text{m}$.

Table 2: Execution time, and relative error for different grid point for $N = 64$ with parameters $k_{on} = 10^{-0.5} \text{s}^{-1}$, $k_{off} = 10^{-1} \text{s}^{-1}$, $D_1 = 30 \mu\text{m}^2/\text{s}$, $D_2 = 10^{-4} \mu\text{m}^2/\text{s}$ and $\omega_m = 0.5 \mu\text{m}$. Table 2 shows the execution time and relative error for different time step sizes, with $N = 64$, for the same parameters.

<table>
<thead>
<tr>
<th>nsteps</th>
<th>time steps=100</th>
<th>time steps=1000</th>
<th>time steps=10000</th>
</tr>
</thead>
<tbody>
<tr>
<td>Execution time (s)</td>
<td>0.79309</td>
<td>6.8334</td>
<td>75.9164</td>
</tr>
<tr>
<td>Relative error</td>
<td>7.0045e-06</td>
<td>6.9974e-07</td>
<td>6.9968e-08</td>
</tr>
</tbody>
</table>

shows first-order accuracy in time, as expected. Table 1 shows the execution time and relative error for different grid sizes for nsteps = 10,000 with parameters $k_{on} = 10^{-0.5} \text{s}^{-1}$, $k_{off} = 10^{-1} \text{s}^{-1}$, $D_1 = 30 \mu\text{m}^2/\text{s}$, $D_2 = 10^{-4} \mu\text{m}^2/\text{s}$ and $\omega_m = 0.5 \mu\text{m}$. Table 2 shows the execution time and relative error for different time step sizes, with $N = 64$, for the same parameters.

Table 3: Execution time and relative error for different grid point for nsteps = 10,000 with parameters $k_{on} = 10^{-2} \text{s}^{-1}$, $k_{off} = 10 \text{s}^{-1}$, $D_1 = 30 \mu\text{m}^2/\text{s}$, $D_2 = 10^{-4} \mu\text{m}^2/\text{s}$ and $\omega_m = 0.5 \mu\text{m}$.

<table>
<thead>
<tr>
<th>N</th>
<th>Execution time (s)</th>
<th>Relative error</th>
</tr>
</thead>
<tbody>
<tr>
<td>8</td>
<td>4.6127</td>
<td>0.0021</td>
</tr>
<tr>
<td>16</td>
<td>7.9169</td>
<td>1.8470e-05</td>
</tr>
<tr>
<td>32</td>
<td>19.9402</td>
<td>2.4014e-06</td>
</tr>
<tr>
<td>64</td>
<td>70.3547</td>
<td>7.8543e-11</td>
</tr>
</tbody>
</table>

For the second case, pure diffusion parameter values are set to be $k_{on} = 10^{-2} \text{s}^{-1}$, $k_{off} = 10 \text{s}^{-1}$, $D_1 = 30 \mu\text{m}^2/\text{s}$, $D_2 = 10^{-4} \mu\text{m}^2/\text{s}$ and $\omega_m = 0.5 \mu\text{m}$ for Figure 2. Figure 2a shows relative error versus grid points per dimension (N) for nsteps = 10,000 and $t_f = 1$. It shows a rapidly decreasing trend for relative error with increasing $N$. Figure 2b shows relative error versus number of time steps for $N = 64$ and $t_f = 1$. As before, first-order accuracy in time is obtained. Table 3 shows the execution time and relative error for different grid point for nsteps = 10,000 with parameters $k_{on} = 10^{-2} \text{s}^{-1}$, $k_{off} = 10 \text{s}^{-1}$, $D_1 = 30 \mu\text{m}^2/\text{s}$, $D_2 = 10^{-4} \mu\text{m}^2/\text{s}$, $\omega_m = 0.5 \mu\text{m}$. Table 4 shows the execution time and relative error for different time steps, with $N = 64$, for the same parameters.

For the third case, full model parameter values are set to be $k_{on} = 10^2 \text{s}^{-1}$, $k_{off} = 10^{-1}$
Figure 2: a: Relative error versus grid points ($N$). b: Relative error versus time steps. Parameters are $k_{on} = 10^{-2} \text{s}^{-1}$, $k_{off} = 10^{-1} \text{s}^{-1}$, $D_1 = 30 \mu\text{m}^2/\text{s}$, $D_2 = 10^{-4} \mu\text{m}^2/\text{s}$ and $\omega_{rn} = 0.5 \mu\text{m}$.

Table 4: Execution time and relative error for different grid point for $N = 64$ with parameters $k_{on} = 10^{-2} \text{s}^{-1}$, $k_{off} = 10^{-1} \text{s}^{-1}$, $D_1 = 30 \mu\text{m}^2/\text{s}$, $D_2 = 10^{-4} \mu\text{m}^2/\text{s}$ and $\omega_{rn} = 0.5 \mu\text{m}$.

<table>
<thead>
<tr>
<th>Execution time(s)</th>
<th>Relative error</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.81812</td>
<td>1.0130e-08</td>
</tr>
<tr>
<td>6.9748</td>
<td>8.6496e-10</td>
</tr>
<tr>
<td>70.3547</td>
<td>7.8543e-11</td>
</tr>
</tbody>
</table>

$s^{-1}$, $D_1 = 30 \mu\text{m}^2/\text{s}$, $D_2 = 10^{-4} \mu\text{m}^2/\text{s}$ and $\omega_{rn} = 0.5 \mu\text{m}$ for Figure 3. Here we consider a diffusion-dominated case. Figure 3 shows relative error versus grid points per dimension ($N$) for $nsteps = 10,000$ and $t_f = 1$. As in the previous cases, it shows a decreasing trend for relative error by increasing $N$. Figure 3b shows relative error versus time steps for $N = 64$ and $t_f = 1$. First-order accuracy in time is again observed. Table 5 shows the execution time and relative error for different grid sizes with $nsteps = 10,000$ and parameters $k_{on} = 10^2 \text{s}^{-1}$, $k_{off} = 10^{-1} \text{s}^{-1}$, $D_1 = 30 \mu\text{m}^2/\text{s}$, $D_2 = 10^{-4} \mu\text{m}^2/\text{s}$ and $\omega_{rn} = 0.5 \mu\text{m}$. Table 6 shows the execution time and relative error for different time step sizes with $N = 64$ for the same parameters.

It is particularly interesting to note that these test cases, the Courant-Friedrichs-Lewy (CFL) condition for forward Euler with $N = 64$ would require $10^5$ time steps to ensure stability, but this KSS method, an explicit method, is able to not only ensure stability but also high accuracy even when greatly exceeding this CFL limit.

6.2. Performance Comparison

In this section we compare the performance of KSS with various other time-stepping methods: forward Euler, fourth-order Runge-Kutta, and Crank-Nicholson. For all methods, the standard 5-point finite difference stencil is used for the Laplacian. The results are shown in Table 7. For both KSS and Crank-Nicholson, only one time step is necessary to obtain a reasonably accurate solution, and we see that the accuracy of the two methods is comparable. However, KSS is significantly faster, and this advantage increases with the number of grid points $N$. Because forward Euler and fourth-order Runge-Kutta are explicit, a very small time step is required to obtain a viable solution, resulting in these methods being literally thousands of times slower than KSS.
Figure 3: a: Relative error versus grid points (N). b: Relative error versus time steps. Parameters are $k_{on} = 10^2 \text{s}^{-1}$, $k_{off} = 10 \text{s}^{-1}$, $D_1 = 30 \mu \text{m}^2/\text{s}$, $D_2 = 10^{-1} \mu \text{m}^2/\text{s}$ and $\omega_{rn} = 0.5 \mu \text{m}$.

Table 5: Execution time and relative error for different grid point for $nsteps = 10,000$ with parameters $k_{on} = 10^2 \text{s}^{-1}$, $k_{off} = 10^{-1} \text{s}^{-1}$, $D_1 = 30 \mu \text{m}^2/\text{s}$, $D_2 = 10^{-1} \mu \text{m}^2/\text{s}$ and $\omega_{rn} = 0.5 \mu \text{m}$.

<table>
<thead>
<tr>
<th>N</th>
<th>Execution time (s)</th>
<th>Relative error</th>
</tr>
</thead>
<tbody>
<tr>
<td>8</td>
<td>4.5256</td>
<td>0.0100</td>
</tr>
<tr>
<td>16</td>
<td>8.0008</td>
<td>9.6420e-04</td>
</tr>
<tr>
<td>32</td>
<td>20.7176</td>
<td>1.7973e-04</td>
</tr>
<tr>
<td>64</td>
<td>76.6904</td>
<td>3.0811e-08</td>
</tr>
</tbody>
</table>

For KSS, two optimizations were made. First, based on observations that the off-diagonal entries of $M_1$ and $\tilde{M}_1$ were, for most frequencies [17], negligibly small compared to the diagonal entries, only the diagonal entries were computed, and then used as the nodes. Second, the formulas in Section 5 were simplified to account for the fact that the components of the initial data are constant functions, which would mean that $\hat{u}$ and $\hat{b}$ would have only one nonzero component.

6.3. Changing Boundary Conditions

We now explain how we can solve (2) using homogeneous Neumann boundary conditions, rather than periodic boundary conditions. The following modifications are needed:

- Using discrete cosine transforms instead of the FFT
- Using frequency ranges $\omega_i = k/2$ for $i = 1, 2$ and $k = 0, 1, 2, \ldots, N - 1$
- Using the appropriate finite difference discretization of the Laplacian, if applicable.

The formulas derived in Section 4 can still be used, for the appropriate values of $\omega$, as the resulting quadrature nodes will have the necessary asymptotic behavior, as demonstrated in [4]. Table 8 shows the results of repeating the experiments of Section 6.2, but with Neumann boundary conditions. We see that all methods exhibit similar accuracy and efficiency.

7. Conclusion

We applied a first-order KSS method to solve the first-order photobleaching kinetics partial differential equations with general initial conditions and the initial conditions that came from a
Table 6: Execution time and relative error for different grid point for $N = 64$ with parameters $k_{in} = 10^2 \text{s}^{-1}$, $k_{off} = 10^{-1} \text{s}^{-1}$, $D_1 = 30 \text{µm}^2/\text{s}$, $D_2 = 10^{-1} \text{µm}^2/\text{s}$ and $\omega_r = 0.5 \text{µm}$.

<table>
<thead>
<tr>
<th>time steps=100</th>
<th>time steps=1000</th>
<th>time steps=10000</th>
</tr>
</thead>
<tbody>
<tr>
<td>Execution time (s)</td>
<td>0.78111</td>
<td>7.6507</td>
</tr>
<tr>
<td>Relative error</td>
<td>3.0821e-06</td>
<td>3.0810e-07</td>
</tr>
</tbody>
</table>

Periodic boundary conditions, reaction dominant case

<table>
<thead>
<tr>
<th>$N = 32$</th>
<th>$N = 64$</th>
</tr>
</thead>
<tbody>
<tr>
<td>time steps</td>
<td>Execution time (s)</td>
</tr>
<tr>
<td>KSS 1</td>
<td>0.0018</td>
</tr>
<tr>
<td>CN 1</td>
<td>0.0131</td>
</tr>
<tr>
<td>rk4 10,000</td>
<td>1.5401</td>
</tr>
<tr>
<td>Euler 100,000</td>
<td>2.3200</td>
</tr>
</tbody>
</table>

Table 7: Execution time and relative error for the solution of (2) using a 1-node KSS method, Crank-Nicholson (CN), fourth-order Runge-Kutta (rk4), and forward Euler, for the reaction-dominant case from Table 1.

pre-bleach steady state. It has been shown that by applying block Arnoldi iteration symbolically for each Fourier coefficient, an approximate analytical solution can be obtained that facilitates qualitative analysis of short-time behavior, which is relevant to the photobleaching stage. The numerical results indicate satisfactory accuracy of the method for all cases, which is promising for application to FRAP laboratory research. We present an approximate analytical solution to this model which makes qualitative analysis feasible for scientists in the field of cell biology.

Future work will consist of proving stability and convergence, consideration of more general laser profiles, and other generalizations of interest, including to three-dimensional problems as in [1]. In [15] KSS was applied to Maxwell’s equations in 3-D; application to a 3-D version of (2) would be analogous. In fact, the approach described in this paper can be used without modification to the formulas herein, except with appropriately adjusted definitions of $\omega$, $L_N$, and $I_r$. Efficient application of higher-order KSS methods (that is, $K > 1$) will also be investigated. Also of interest are more general models in which the coefficients vary over time, as would be the case for modeling both photobleaching and recovery, for either a system of linear or nonlinear PDEs. For this case, a combination of KSS and EPI methods [19] introduced in [4] can be applied, in which KSS methods are used to approximate required matrix function-vector products.
Neumann boundary conditions, reaction dominant case

<table>
<thead>
<tr>
<th></th>
<th>$N = 32$</th>
<th>$N = 64$</th>
</tr>
</thead>
<tbody>
<tr>
<td>time steps</td>
<td>Execution time (s)</td>
<td>Rel error</td>
</tr>
<tr>
<td>KSS</td>
<td>1</td>
<td>0.0014</td>
</tr>
<tr>
<td>CN</td>
<td>1</td>
<td>0.0127</td>
</tr>
<tr>
<td>rk4</td>
<td>10,000</td>
<td>1.5437</td>
</tr>
<tr>
<td>Euler</td>
<td>100,000</td>
<td>3.0507</td>
</tr>
</tbody>
</table>

Table 8: Execution time and relative error for the solution of (2) using a 1-node KSS method, Crank-Nicholson (CN), fourth-order Runge-Kutta (rk4), and forward Euler, for the reaction-dominant case from Table 1, except with homogeneous Neumann boundary conditions.

Appendix A. Appendix

Appendix A.1. Block Arnoldi, $\omega \neq 0$

Here, we present formulas for the entries of $M_1$ and $\tilde{M}_1$ from equations (21) and (22), respectively. We first define

\[
\begin{align*}
T_1 &= v_1^2(\omega)\hat{u} + v_{21}(\omega)v_{11}(\omega)\hat{b} \\
T_2 &= v_2^2(\omega)\hat{b} + v_{21}(\omega)v_{11}(\omega)\hat{u} \\
T_3 &= v_1^2(\omega)\overline{\hat{u}} + v_{21}(\omega)v_{11}(\omega)\overline{\hat{b}} \\
T_4 &= v_2^2(\omega)\overline{\hat{b}} + v_{21}(\omega)v_{11}(\omega)\overline{\hat{u}} \\
T_5 &= v_{12}^2(\omega)\hat{u} + v_{22}(\omega)v_{12}(\omega)\hat{b} \\
T_6 &= v_{22}^2(\omega)\hat{b} + v_{22}(\omega)v_{12}(\omega)\hat{u} \\
T_7 &= v_{12}^2(\omega)\overline{\hat{u}} + v_{22}(\omega)v_{12}(\omega)\overline{\hat{b}} \\
T_8 &= v_{22}^2(\omega)\overline{\hat{b}} + v_{22}(\omega)v_{12}(\omega)\overline{\hat{u}}
\end{align*}
\]

Then, for $M_1$, we have

\[
\begin{align*}
M_{11} &= -k_b T_{r_1} + \frac{n_1}{\|v_1(\omega)\|^2} \\
M_{12} &= \frac{n_2}{d_1} \\
M_{21} &= \frac{n_3}{d_1} \\
M_{22} &= \frac{n_4}{d_2}
\end{align*}
\]

20
while for $\ddot{M}_1$, we have

$$\ddot{M}_{11} = -k_b \overline{r_n} + \frac{n_5}{\|v_2(\omega)\|^2}$$

$$\ddot{M}_{12} = \frac{n_6}{d_3}$$

$$\ddot{M}_{21} = \frac{n_7}{d_3}$$

$$\ddot{M}_{22} = \frac{n_8}{d_4}$$

where

$$n_1 = k_{\text{off}}v_{111}(\omega)v_{21}(\omega) + k_{\text{an}}v_{213}(\omega)v_{11}(\omega) - (D_1v_{11}^2(\omega) + D_2v_{21}^2(\omega))[|\omega|^2 - k_{\text{off}}v_{11}^2(\omega) - k_{\text{an}}v_{11}^2(\omega)]$$

$$n_2 = -k_b||v_1(\omega)||^2(I_{n\text{,off}}v_{111}(\omega) + I_{n\text{,an}}b v_{21}(\omega)) + k_n\overline{r_n}[T_1v_{11}(\omega) + T_2v_{21}(\omega)] - ||\omega||^2||v_1(\omega)||^2(D_1\overline{v}_{11}(\omega)\overline{u} + D_2v_{213}(\omega)\overline{b}) + ||\omega||^2[D_1\overline{v}_{11}(\omega)T_1 + D_2v_{213}(\omega)T_2] + ||v_1(\omega)||^2(v_{11}(\omega)(k_{\text{off}}\overline{b} - k_{\text{an}}\overline{u}) + v_{21}(\omega)(k_{\text{an}}\overline{u} - k_{\text{off}}\overline{b})) + (v_{11}(\omega) - v_{21}(\omega))\{k_{\text{off}}T_1 - k_{\text{off}}T_2]\}$$

$$n_3 = -k_b||v_1(\omega)||^2(I_{n\text{,off}}v_{111}(\omega) + I_{n\text{,an}}b v_{21}(\omega)) + k_n\overline{r_n}[T_3v_{11}(\omega) + T_4v_{21}(\omega)] - ||\omega||^2||v_1(\omega)||^2((D_1v_{11}(\omega)\overline{u} + D_2v_{21}(\omega)\overline{b}) + ||\omega||^2[D_1v_{11}(\omega)T_3 + D_2v_{21}(\omega)T_4] + ||v_1(\omega)||^2(v_{11}(\omega)(k_{\text{off}}\overline{b} - k_{\text{an}}\overline{u}) + v_{21}(\omega)(k_{\text{an}}\overline{u} - k_{\text{off}}\overline{b})) + (v_{11}(\omega)k_{\text{an}} - v_{21}(\omega)k_{\text{off}})[T_3 - T_4]\}$$

$$d_1 = \sqrt{||v_1(\omega)||^2} \times$$

$$N \sqrt{||v_1(\omega)||^2(|\omega|^2 + |b|^2) - \text{Real}(2||v_1(\omega)||^2[\overline{u}T_1 + \overline{b}T_2]) + [(T_2)^2 + (T_1)^2]}$$

$$d_2 = N^2||v_1(\omega)||^2(|\omega|^2 + |b|^2) - \text{Real}(2||v_1(\omega)||^2[\overline{u}T_1 + \overline{b}T_2]) + [(T_2)^2 + (T_1)^2]$$

21
\[ n_4 = [-k_{\text{off}}(T_2)^2 - k_{\text{on}}(T_1)^2] + k_{\text{on}}T_1T_4 + k_{\text{off}}T_3T_2 - N^2k_6|v_1(\omega)|^4(\bar{I}_u, \tilde{u}^2 + \bar{I}_v, \tilde{b}^2) + k_6|v_1(\omega)|^2[\bar{I}_u, \tilde{u}T_3 + \bar{I}_v, \tilde{b}T_4] - N^2|v_1(\omega)|^4(D_1||\nabla u||^2 + D_2||\nabla b||^2) - ||v_1(\omega)||^2T_1[-k_{\text{off}}\bar{b} + D_1||\omega||^2\bar{u} + k_{\text{on}}\tilde{u}] + ||v_1(\omega)||^2T_2[-k_{\text{off}}\tilde{u} + D_2||\omega||^2\bar{b} + k_{\text{off}}\tilde{b}] - k_6\bar{I}_T[(T_2)^2 + (T_5)^2] + (k_{\text{off}}\tilde{b} - k_{\text{on}}\tilde{u})||v_1(\omega)||^2T_4 - ||\omega||^2[D_1(T_3)^2 + D_2(T_3)^2] + (k_{\text{on}}\tilde{u} - k_{\text{off}}\tilde{b})||v_1(\omega)||^2T_3 + N^2|v_1(\omega)|^4[-k_{\text{on}}||u||^2 + N^2k_{\text{off}}\tilde{u}\bar{b} + N^2k_{\text{on}}\tilde{u}\bar{b} - k_{\text{off}}||b||^2] - ||\omega||^2|v_1(\omega)|^2[D_1\tilde{u}T_3 + D_2\tilde{b}T_4] + k_6|v_1(\omega)|^2[\bar{I}_u, \tilde{b}T_2 + \bar{I}_v, \tilde{u}T_1] \]

\[ n_5 = k_{\text{off}}v_1(\omega)v_2(\omega) + k_{\text{on}}v_2(\omega)v_1(\omega) - (D_1v_1(\omega) + D_2v_2(\omega))||\omega||^2 - k_{\text{off}}v_2(\omega) - k_{\text{on}}v_1(\omega) \]

\[ n_6 = -k_6|v_2(\omega)|^2(\bar{I}_u, \tilde{u}v_1(\omega) + \bar{I}_v, \tilde{b}v_2(\omega)) + k_6\bar{I}_T[T_5v_1(\omega) + T_6v_2(\omega)] - ||\omega||^2|v_2(\omega)|^2(D_1v_1(\omega)\bar{u} + D_2v_2(\omega)\tilde{b}) + ||\omega||^2[D_1v_1(\omega)T_5 + D_2v_2(\omega)T_6] + ||v_2(\omega)||^2(v_1(\omega)(k_{\text{off}}\tilde{b} - k_{\text{on}}\tilde{u}) + v_2(\omega)(k_{\text{on}}\tilde{u} - k_{\text{off}}\tilde{b})) + (v_1(\omega) - v_2(\omega))[k_{\text{on}}T_5 - k_{\text{off}}T_6] \]

\[ n_7 = -k_6|v_2(\omega)|^2[\bar{I}_u, \tilde{u}v_1(\omega) + \bar{I}_v, \tilde{b}v_2(\omega)] + k_6\bar{I}_T[T_7v_1(\omega) + T_8v_2(\omega)] - ||\omega||^2|v_2(\omega)|^2(D_1v_1(\omega)\bar{u} + D_2v_2(\omega)\tilde{b}) + ||\omega||^2[D_1v_1(\omega)T_7 + D_2v_2(\omega)T_8] + ||v_2(\omega)||^2(v_1(\omega)(k_{\text{off}}\tilde{b} - k_{\text{on}}\tilde{u}) + v_2(\omega)(k_{\text{on}}\tilde{u} - k_{\text{off}}\tilde{b})) + (v_1(\omega)k_{\text{off}} - v_2(\omega)k_{\text{on}})[T_7 - T_8] \]

\[ d_3 = \sqrt{|v_2(\omega)|^2} \times \left( N^2|v_2(\omega)|^2(||u^2|| + ||b^2||) - \text{Real}(2|v_2(\omega)|^2[\bar{u}\tilde{u}T_5 + \bar{b}\tilde{b}T_6]) + (|T_6|^2 + (T_5)^2) \right) \]

\[ d_4 = N^2|v_2(\omega)|^2(||u^2|| + ||b^2||) - \text{Real}(2|v_2(\omega)|^2[\bar{u}\tilde{u}T_5 + \bar{b}\tilde{b}T_6]) + (|T_6|^2 + (T_5)^2) \]

\[ n_8 = [-k_{\text{off}}(T_6)^2 - k_{\text{on}}(T_5)^2] + k_{\text{on}}T_5T_8 + k_{\text{off}}T_7T_6 - N^2k_6|v_2(\omega)|^4(\bar{I}_u, \tilde{u}^2 + \bar{I}_v, \tilde{b}^2) + k_6|v_2(\omega)|^2[\bar{I}_u, \tilde{u}T_7 + \bar{I}_v, \tilde{b}T_8] - N^2|v_2(\omega)|^4(D_1||\nabla u||^2 + D_2||\nabla b||^2) - ||v_2(\omega)||^2T_5[-k_{\text{off}}\bar{b} + D_1||\omega||^2\bar{u} + k_{\text{on}}\tilde{u}] + ||v_2(\omega)||^2T_6[-k_{\text{off}}\tilde{u} + D_2||\omega||^2\bar{b} + k_{\text{off}}\tilde{b}] - k_6\bar{I}_T[(T_6)^2 + (T_5)^2] + (k_{\text{off}}\tilde{b} - k_{\text{on}}\tilde{u})v_2(\omega)||v_2(\omega)||^2T_8 - ||\omega||^2[D_1(T_7)^2 + D_2(T_5)^2] + (k_{\text{on}}\tilde{u} - k_{\text{off}}\tilde{b})||v_2(\omega)||^2T_7 + N^2|v_2(\omega)|^4[-k_{\text{on}}||u||^2 + N^2k_{\text{off}}\tilde{u}\bar{b} + N^2k_{\text{on}}\tilde{u}\bar{b} - k_{\text{off}}||b||^2] - ||\omega||^2|v_2(\omega)|^2[D_1\tilde{u}T_7 + D_2\tilde{b}T_8] + k_6|v_2(\omega)|^2[\bar{I}_u, \tilde{b}T_6 + \bar{I}_v, \tilde{u}T_5]. \]

**Appendix A.2. Block Arnoldi, \omega_1 = \omega_2 = 0**

The entries of \( M_1 \) from Section 3.3 are given by

\[
M_1 = \begin{bmatrix} X_1 \end{bmatrix} L X_1
= \begin{bmatrix} M_{11} & M_{12} \\ M_{21} & M_{22} \end{bmatrix}
\]
where

\[ M_{11} = -T_n k_b \]
\[ M_{12} = -N k_b (u T_n + b T_n + T_n (\bar{u} + \bar{b})) \]
\[ M_{21} = \frac{N ((k_{\text{off}} - k_{\text{on}}) (\bar{u} - \bar{b}) - k_b (u T_n + b T_n + T_n (\bar{u} + \bar{b})))}{\sqrt{2 d_1}} \]
\[ M_{22} = \frac{n_1}{d_1} \]

with

\[ n_1 = -k_b N^2 \left( T_n u^2 + T_n b^2 - (\bar{u} + \bar{b}) T_n u - (\bar{u} + \bar{b}) T_n b + T_n \left( \frac{\bar{u}^2}{2} + \frac{\bar{b}^2}{2} + \bar{u} \bar{b} \right) \right) - k_{\text{on}} \left( \|u\|^2 - N^2 \bar{u} (\bar{u} + \bar{b}) + \frac{N^2 (\bar{b} + \bar{u})^2}{4} \right) - k_{\text{off}} \left( \|b\|^2 - N^2 \bar{b} (\bar{u} + \bar{b}) + \frac{N^2 (\bar{b} + \bar{u})^2}{4} \right) + N^2 (k_{\text{on}} + k_{\text{off}}) \left( \frac{\bar{u} b - \frac{\bar{u}^2}{2} - \frac{\bar{b}^2}{2}}{4} - \frac{\bar{b} \bar{u}}{2} - D_1 \|\nabla u\|^2 + D_2 \|\nabla b\|^2 \right) \]
\[ d_1 = \|u\|^2 + \|b\|^2 - \frac{N^2 (\bar{b} + \bar{u})^2}{2}. \]

Similarly, the entries of \( \tilde{M}_1 \) from Section 3.3 are

\[ \tilde{M}_1 = \tilde{X}_1^T L \tilde{X}_1 \]
\[ = \begin{bmatrix} \tilde{M}_{11} & \tilde{M}_{12} \\ \tilde{M}_{21} & \tilde{M}_{22} \end{bmatrix} \]

where

\[ \tilde{M}_{11} = -T_n k_b - k_{\text{on}} - k_{\text{off}} \]
\[ \tilde{M}_{12} = \frac{k_b N (-k_{\text{on}} T_n u + k_{\text{off}} T_n b + T_n (\bar{u} k_{\text{on}} - \bar{b} k_{\text{off}})))}{\sqrt{(k_{\text{on}}^2 + k_{\text{off}}^2) d_2}} \]
\[ \tilde{M}_{21} = \tilde{M}_{12} + \frac{N \left( \bar{u} \left( -k_{\text{off}}^2 + \frac{k_{\text{off}}^2}{k_{\text{on}}^2 + k_{\text{off}}^2} \right) + \bar{b} \left( k_{\text{on}}^2 - \frac{k_{\text{off}}^2}{k_{\text{on}}^2 + k_{\text{off}}^2} \right) \right)}{\sqrt{(k_{\text{on}}^2 + k_{\text{off}}^2) d_2}} \]
\[ \tilde{M}_{22} = \frac{n_2}{d_2} \]
with

\[ d_2 = ||a||^2 + ||b||^2 - \frac{N^2 \tilde{u}_n^2 k_{on}^2}{k_{on}^2 + k_{off}^2} - \frac{N^2 \tilde{f}_n^2 k_{off}^2}{k_{on}^2 + k_{off}^2} + \frac{2N^2 \tilde{u}_b k_{off}}{k_{on}^2 + k_{off}^2} \]

\[ n_2 = -k_b N^2 \left( I_{\nu} u^2 + I_{\nu} b^2 + I_{\nu} u \left( \frac{2 \tilde{u}_n^2}{k_{on}^2 + k_{off}^2} - \frac{2 \tilde{b}_n k_{off}}{k_{on}^2 + k_{off}^2} \right) \right) + \frac{2 \tilde{b}_n k_{off}}{k_{on}^2 + k_{off}^2} \]

\[ N^2 \left( \tilde{u}_b (k_{on} + k_{off}) + \tilde{b}_n (k_{on} + k_{off}) - \tilde{b}_n (k_{on} + k_{off}) \right) - (k_{on} ||a||^2 + k_{off} ||b||^2) - (D_1 ||\nabla a||^2 + D_2 ||\nabla b||^2). \]

**Appendix A.3. Approximate Analytical Solution, \( \omega \neq 0 \) Case**

The entries of \( M_1 \) and \( \tilde{M}_1 \) from Section 4.1 are given by

\[ M_1 = X^H \cdot L \cdot X \]

\[ = \begin{bmatrix}
\frac{n_3}{\nu_{22}^2 + k_{off}^2} & \frac{k_b k_{off} I_{\nu} (\nu_{22} - k_{on})}{N^2 \sqrt{\nu_{22}^2 + k_{off}^2} \sqrt{k_{on}^2 + k_{off}^2}} \\
\frac{k_b k_{off} I_{\nu} (\nu_{22} - k_{on})}{N^2 \sqrt{\nu_{22}^2 + k_{off}^2} \sqrt{k_{on}^2 + k_{off}^2}} & -k_b I_{\nu}
\end{bmatrix} \]

\[ n_4 = -k_b I_{\nu} (\nu_{22}^2 + k_{off}^2) - ||a||^2 (D_1 \nu_{22}^2 + k_{off}^2) - (k_{on} \nu_{22}^2 + \nu_{22} k_{off}^2 + k_{off} k_{on} u_{22} + k_{off}^3) \]

\[ \tilde{M}_1 = \tilde{X}^H \cdot L \cdot \tilde{X} \]

\[ = \begin{bmatrix}
\frac{n_5}{\nu_{21}^2 + k_{off}^2} & \frac{k_b k_{off} I_{\nu} (-\nu_{21} + k_{on})}{N^2 \sqrt{\nu_{21}^2 + k_{off}^2} \sqrt{k_{on}^2 + k_{off}^2}} \\
\frac{k_b k_{off} I_{\nu} (-\nu_{21} + k_{on})}{N^2 \sqrt{\nu_{21}^2 + k_{off}^2} \sqrt{k_{on}^2 + k_{off}^2}} & -k_b I_{\nu}
\end{bmatrix} \]

\[ n_5 = -k_b I_{\nu} (\nu_{21}^2 + k_{off}^2) - ||a||^2 (D_1 \nu_{21}^2 + k_{off}^2) - (k_{on} \nu_{21}^2 + \nu_{21} k_{off}^2 + k_{off} k_{on} u_{21} + k_{off}^3) \]

(A.1)

where \( \tilde{f} \) is the complex conjugate of the discrete Fourier transform of the grid function \( f \).
The eigenvalues of $M_1$ are

$$\lambda_{1,\omega} = \frac{1}{2(u_{22}^2 + k_{\text{off}}^2)} \left[ \frac{1}{2} \frac{1}{(u_{22}^2 + k_{\text{off}}^2)^2} ||\omega||^4 (D_1 u_{22}^2 + k_{\text{off}}^2 D_2)^2 + \frac{1}{(u_{22}^2 + k_{\text{off}}^2)^2} (k_{\text{on}} u_{22}^2 + \overline{\omega}^2 k_{\text{off}}^2 + k_{\text{off}} k_{\text{on}} u_{22} + k_{\text{off}}^3) + \frac{2}{(u_{22}^2 + k_{\text{off}}^2)^2} ||\omega||^2 (D_1 u_{22}^2 + k_{\text{off}}^2 D_2)(k_{\text{on}} u_{22}^2 + \overline{\omega}^2 k_{\text{off}}^2 + k_{\text{off}} k_{\text{on}} u_{22} + k_{\text{off}}^3) + 4k_{\text{off}}^2 k_{\text{on}}^2 \overline{\omega}^2 \left( u_{22} - k_{\text{on}} \right)(u_{22} - k_{\text{on}}) \right]^{1/2}
$$

$$\lambda_{2,\omega} = \frac{n_6}{d_6}
$$

where

$$d_6 = \frac{-1}{u_{22}^2 + k_{\text{off}}^2} \left[ ||\omega||^2 (D_1 u_{22}^2 + k_{\text{off}}^2 D_2) + (k_{\text{on}} u_{22}^2 + \overline{\omega}^2 k_{\text{off}}^2 + k_{\text{off}} k_{\text{on}} u_{22} + k_{\text{off}}^3) \right] - 2k_{\text{on}} T_r + \frac{1}{(u_{22}^2 + k_{\text{off}}^2)^2} ||\omega||^4 (D_1 u_{22}^2 + k_{\text{off}}^2 D_2)^2 + \frac{1}{(u_{22}^2 + k_{\text{off}}^2)^2} (k_{\text{on}} u_{22}^2 + \overline{\omega}^2 k_{\text{off}}^2 + k_{\text{off}} k_{\text{on}} u_{22} + k_{\text{off}}^3)^2 + \frac{2}{(u_{22}^2 + k_{\text{off}}^2)^2} ||\omega||^2 (D_1 u_{22}^2 + k_{\text{off}}^2 D_2)(k_{\text{on}} u_{22}^2 + \overline{\omega}^2 k_{\text{off}}^2 + k_{\text{off}} k_{\text{on}} u_{22} + k_{\text{off}}^3) + 4k_{\text{off}}^2 k_{\text{on}}^2 \overline{\omega}^2 \left( u_{22} - k_{\text{on}} \right)(u_{22} - k_{\text{on}}) \right]^{1/2}
$$

$$n_6 = \frac{2}{(u_{22}^2 + k_{\text{off}}^2)} \left[ k_{\text{on}} T_r ||\omega||^2 (D_1 u_{22}^2 + k_{\text{off}}^2 D_2) + k_{\text{off}}^2 T_r^2 (u_{22}^2 + k_{\text{off}}^2) + k_{\text{off}} T_r (k_{\text{on}} u_{22}^2 + \overline{\omega}^2 k_{\text{off}}^2 + k_{\text{off}} k_{\text{on}} u_{22} + k_{\text{off}}^3) - 2k_{\text{off}}^2 k_{\text{on}}^2 \overline{\omega}^2 \left( u_{22} - k_{\text{on}} \right)(u_{22} - k_{\text{on}}) \right] - \frac{2k_{\text{off}}^2 k_{\text{on}}^2 \overline{\omega}^2 \left( u_{22} - k_{\text{on}} \right)(u_{22} - k_{\text{on}})}{N^4 (k_{\text{on}}^2 + k_{\text{off}}^2)(k_{\text{off}}^2 + u_{22}^2)}.\quad (A.2)
Similarly, the eigenvalues of $\tilde{M}_1$ are

$$\lambda_{1,\omega} = -\frac{1}{2(u_{21}^2 + k_{\text{off}}^2)} \left( ||\omega||^2(D_1 u_{21}^2 + k_{\text{off}}^2 D_2) + (k_{\text{on}} u_{21}^2 + u_{21}^3 k_{\text{off}}^2 + k_{\text{off}} k_{\text{on}} u_{21} + k_{\text{off}}^3) \right) - \frac{1}{2(u_{21}^2 + k_{\text{off}}^2)} \left[ k_{b} T_\omega - \frac{1}{(u_{21}^2 + k_{\text{off}}^2)^2} ||\omega||^4(D_1 u_{21}^2 + k_{\text{off}}^2 D_2)^2 + \right. $$

$$\left. \frac{1}{(u_{21}^2 + k_{\text{off}}^2)^2} (k_{\text{on}} u_{21}^2 + u_{21} k_{\text{off}} + k_{\text{off}} k_{\text{on}} u_{21} + k_{\text{off}}^3)^2 + \right. $$

$$\left. \frac{2}{(u_{21}^2 + k_{\text{off}}^2)^2} ||\omega||^2(D_1 u_{21}^2 + k_{\text{off}}^2 D_2)(k_{\text{on}} u_{21}^2 + u_{21} k_{\text{off}} + k_{\text{off}} k_{\text{on}} u_{21} + k_{\text{off}}^3) \right]$$

$$\frac{n_7}{d_7} \quad \frac{4 k_{\text{off}}^2 P_r (\mu_{21} + k_{\text{on}})(-u_{21} + k_{\text{on}})^{1/2}}{N^4(k_{\text{on}} + k_{\text{off}})(k_{\text{off}}^2 + u_{21}^2)}$$

where

$$d_7 = \frac{-1}{u_{21}^2 + k_{\text{off}}^2} \left[ ||\omega||^2(D_1 u_{21}^2 + k_{\text{off}}^2 D_2) + (k_{\text{on}} u_{21}^2 + u_{21} k_{\text{off}} + k_{\text{off}} k_{\text{on}} u_{21} + k_{\text{off}}^3) \right] - $$

$$2k_{b} T_\omega - \frac{1}{(u_{21}^2 + k_{\text{off}}^2)^2} ||\omega||^4(D_1 u_{21}^2 + k_{\text{off}}^2 D_2)^2 + $$

$$\frac{1}{(u_{21}^2 + k_{\text{off}}^2)^2} (k_{\text{on}} u_{21}^2 + u_{21} k_{\text{off}} + k_{\text{off}} k_{\text{on}} u_{21} + k_{\text{off}}^3)^2 + $$

$$\frac{2}{(u_{21}^2 + k_{\text{off}}^2)^2} ||\omega||^2(D_1 u_{21}^2 + k_{\text{off}}^2 D_2)(k_{\text{on}} u_{21}^2 + u_{21} k_{\text{off}} + k_{\text{off}} k_{\text{on}} u_{21} + k_{\text{off}}^3) \right]$$

$$\frac{n_7}{d_7} \quad \frac{4 k_{\text{off}}^2 P_r (\mu_{21} + k_{\text{on}})(-u_{21} + k_{\text{on}})^{1/2}}{N^4(k_{\text{on}} + k_{\text{off}})(k_{\text{off}}^2 + u_{21}^2)}$$

We note that $\tilde{P}_r$ decays rapidly to zero at higher frequencies, so for such components it can be neglected for simplicity.


