Available online at www.sciencedirect.com
ScienceDirect
Journal of Computational Physics 226 (2007) 1509–1531
JOURNAL OF COMPUTATIONAL PHYSICS
www.elsevier.com/locate/jcp

A fully adaptive reaction–diffusion integration scheme with applications to systems biology

David J. Miller, Avijit Ghosh *

Department of Physics, Drexel University, 3141 Chestnut Street, Philadelphia, PA 19104, USA

Received 27 June 2006; received in revised form 6 April 2007; accepted 25 May 2007
Available online 12 June 2007

Abstract

Adaptive integration schemes for ODE systems typically function by adjusting the time step size so as to keep the truncation error below some desired value. For adaptive integration of PDE systems involving coupled kinetic reaction and diffusion operations, truncation error arises not only from the individual propagators but also from their method of coupling. A common second-order accurate method for coupling operators is Strang's method of operator splitting. We derive an expression for the truncation error resulting from Strang splitting reaction and diffusion operators for an arbitrary number of spatial dimensions, and demonstrate its use in adaptive time step algorithms. In addition, we present explanations of the second order implicit reaction and diffusion operators, and their individual error calculations used in our implementation of the scheme. Finally, using example simulations we discuss the use of this calculation for problems in systems biology.

Keywords: Adaptive integration; Reaction–diffusion systems; Strang splitting; Systems biology

1. Introduction

Fully-adaptive time integration for multi-dimensional reaction–diffusion PDE problems requires more than a simple error estimate from each integration operator at each time step. Truncation error can also arise from the method in which the operators are applied. A popular second order method for combining operators, particularly in atmospheric science [1], is Strang's method of operator splitting or "Strang splitting" [2–4]. In the first part of this paper, we derive an expression for the truncation error resulting from Strang splitting reaction and diffusion operators to form a second order integration scheme. We then present explanations of the specific implicit reaction and implicit diffusion operators used as well as their individual error calculations. The method is implemented in CellSim [5], a PDE-based cell simulation software package developed by our group and freely available under the GNU Public License [6]. Finally, we give examples of and discuss the use of this calculation for problems in systems biology.

* Corresponding author. Tel.: +1 215 895 2726; fax: +1 215 895 5934.
E-mail addresses: david.jonathan.miller@drexel.edu (D.J. Miller), avijit@physics.drexel.edu (A. Ghosh).

0021-9991/$ - see front matter © 2007 Elsevier Inc. All rights reserved.
doi:10.1016/j.jcp.2007.05.031

2. Strang splitting truncation error

We first derive an expression for the truncation error due to Strang splitting reaction and diffusion operators [1]. Our generalized system consists of a vector of chemical concentrations $\boldsymbol{C}(\vec{x},t)$ which evolve in time and space according to specified differential equations

$$\frac{\partial}{\partial t}\boldsymbol{C}(\vec{x},t) = F(\boldsymbol{C}(\vec{x},t)) = F_{\mathrm{R}}(\boldsymbol{C}(\vec{x},t)) + F_{\mathrm{D}}(\boldsymbol{C}(\vec{x},t)), \tag{1}$$

where the expressions

$$F_{\mathrm{R}}(\boldsymbol{C}(\vec{x},t)) = R(\boldsymbol{C}(\vec{x},t)) = \sum_{i=0}^{n}\left(\alpha_i \prod_{j=0}^{m}[\boldsymbol{C}_j(\vec{x},t)]^{\beta_{ij}}\right), \quad \alpha_i \in \mathbb{R},\ \beta_{ij} \in \mathbb{N}_0, \tag{2}$$

$$F_{\mathrm{D}}(\boldsymbol{C}(\vec{x},t)) = \boldsymbol{D}\nabla^2\boldsymbol{C}(\vec{x},t) \tag{3}$$

are the source term vectors for chemical kinetics and for simple constant-rate diffusion, respectively. Kinetics is composed of a sum over n terms, each of which is a product of a rate constant α_i and concentrations from amongst the system's m reactants, and $\boldsymbol{D}$ is a diagonal matrix of diffusion constants. The exponent β_{ij} determines which reactants contribute to each term. Throughout the paper, we will assume that all operators possess both time and space dependence, and we will specify this dependence when required for clarity. The function space $\mathbf{S}$ consists of all operators of interest that can act on $\boldsymbol{C}$. For reaction–diffusion systems, this consists of $\{I, F_{\mathrm{R}}, F_{\mathrm{D}}\}$ and any linear combination of these.

Following the work of Lanser and Verwer [1] using Lie operator notation [7] adapted from Sanz-Serna and Calvo [8,9], we start with a reaction–diffusion solution operator $\mathcal{S}(\delta t)$ for Eq. (1) that acts on a solution at time t to give a solution at time $t+\delta t$,

$$\boldsymbol{C}(t+\delta t) = \mathcal{S}(\delta t)\boldsymbol{C}(t). \tag{4}$$

Let $\tilde{\mathcal{S}}(\delta t)$ denote a numerical approximation to $\mathcal{S}(\delta t)$, such that $\tilde{\boldsymbol{C}}(t+\delta t) = \tilde{\mathcal{S}}(\delta t)\boldsymbol{C}(t)$.

By combining the reaction and diffusion sub-operators using Strang splitting [2] and using the preferred order of the stiff and non-stiff operators as outlined in [10,11] and further applied in [12,13], the solution operator becomes

$$\tilde{\mathcal{S}}(\delta t) \equiv \tilde{\mathcal{S}}_{\mathrm{R}}\left(\frac{\delta t}{2}\right)\tilde{\mathcal{S}}_{\mathrm{D}}(\delta t)\tilde{\mathcal{S}}_{\mathrm{R}}\left(\frac{\delta t}{2}\right). \tag{5}$$

The truncation error associated with using the approximate solution operator from Eq. (5) can be found as follows: For each operator F_i in our function space $\boldsymbol{S}$, a Lie operator $\mathcal{F}_i$ is associated. This linear operator $\mathcal{F}_i$ maps *any* operator G in $\boldsymbol{S}$ into the operator $\mathcal{F}_i \cdot G$ such that [8]

$$(\mathcal{F}_i \cdot G)(\boldsymbol{C}) = F_i(\boldsymbol{C})\frac{\partial}{\partial \boldsymbol{C}}G(\boldsymbol{C}). \tag{6}$$

We will use this property to find a propagator for the problem at hand. Let F_i be the operator $F = \frac{\partial}{\partial t}$ defined by the left-hand side of Eq. (1). Substituting F into the mapping of Eq. (6) yields

$$(\mathcal{F} \cdot G)(\boldsymbol{C}) = \frac{\partial \boldsymbol{C}}{\partial t}\frac{\partial}{\partial \boldsymbol{C}}G(\boldsymbol{C}), \tag{7}$$

$$= \frac{\partial}{\partial t}G(\boldsymbol{C}). \tag{8}$$

Moreover, by recursively applying Eq. (6) using F we find

$$(\mathcal{F}^k \cdot G)(\boldsymbol{C}) = \frac{\partial^k}{\partial t^k}G(\boldsymbol{C}). \tag{9}$$

Evaluating these derivatives at time $t = 0$ we find the Taylor expansion of G about time t,

$$\left(\sum \frac{\delta t^k \mathcal{F}^k}{k!} \cdot G\right)\boldsymbol{C}(t) = ((\mathrm{e}^{\delta t\mathcal{F}}) \cdot G)\boldsymbol{C}(t). \tag{10}$$

Of particular interest is the case of $G = I$, the identity operator. Substituting it in reveals

$$((\mathrm{e}^{\delta t\mathcal{F}})\cdot I)\boldsymbol{C}(t) = \left(I + \delta t\mathcal{F}I + \frac{\delta t^2}{2}\mathcal{F}^2 I + \cdots\right)\boldsymbol{C}(t), \tag{11}$$

$$= \left(1 + \delta t\frac{\partial}{\partial t} + \frac{\delta t^2}{2}\frac{\partial^2}{\partial t^2} + \cdots\right)\boldsymbol{C}(t), \tag{12}$$

$$= \boldsymbol{C}(t + \delta t), \tag{13}$$

thus propagating the system a step δt forward in time. It is critical to point out that the above procedure spanning Eqs. (7)–(13) applies for all operators in $\boldsymbol{S}$, not just $F = \frac{\partial}{\partial t}$. Consequently, as our Strang-split propagation operator is actually a combination of sub-operators, this treatment applies such that

$$\mathrm{e}^{\delta t\tilde{\mathcal{F}}} \equiv \mathrm{e}^{\frac{\delta t}{2}\mathcal{F}_\mathrm{R}}\mathrm{e}^{\delta t\mathcal{F}_\mathrm{D}}\mathrm{e}^{\frac{\delta t}{2}\mathcal{F}_\mathrm{R}}, \tag{14}$$

where we have performed the above procedure three times, using the appropriate ordering of $\mathcal{F}_\mathrm{R}$ and $\mathcal{F}_\mathrm{D}$, the Lie operator representations of our reaction and diffusion operators, from the splitting scheme in Eq. (5). We note that $\tilde{\mathcal{F}}$ represents a numerical approximation to $\mathcal{F}$.

Since the reaction and diffusion operators do not in general commute, we make use of the BCH formula [14–16] for gauging the splitting error. The BCH formula stipulates that an expression of linear operators X and Y of the form $e^X e^Y$ can be written as e^Z in terms of commutators, where

$$Z = X + Y + \frac{1}{2}[X, Y] + \frac{1}{12}([X, [X, Y]] + [Y, [Y, X]]) + \frac{1}{24}[X, [Y, [Y, X]]] + \cdots \tag{15}$$

Applying this formula twice to Eq. (14) leads to an expression for the exponent:

$$\tilde{\mathcal{F}} = \mathcal{F}_\mathrm{D} + \mathcal{F}_\mathrm{R} + \frac{\delta t^2}{24}[\mathcal{F}_\mathrm{R}, [\mathcal{F}_\mathrm{D}, \mathcal{F}_\mathrm{R}]] + \frac{\delta t^2}{12}[\mathcal{F}_\mathrm{D}, [\mathcal{F}_\mathrm{D}, \mathcal{F}_\mathrm{R}]] + \mathcal{O}(\delta t^4), \tag{16}$$

where we note that odd-ordered terms cancel. The second order and higher terms of Eq. (16) constitute the error due to splitting the exact reaction and diffusion Lie operators $\mathcal{F}_\mathrm{R}$ and $\mathcal{F}_\mathrm{D}$. What we are interested in finding is a similar expression to Eq. (16) but in terms of the numerical operators F_R and F_D,

$$\frac{\tilde{\partial}}{\partial t}\boldsymbol{C}(t) = \tilde{F}(\boldsymbol{C}) \equiv F_\mathrm{R}(\boldsymbol{C}) + F_\mathrm{D}(\boldsymbol{C}) + \delta t^2 E_\mathrm{F}(\boldsymbol{C}) + \mathcal{O}(\delta t^4), \tag{17}$$

thus E_F will reveal the second order splitting error of the propagation operator from Eq. (14). To find this we must first convert E_F from an expression of Lie operators to an expression of numerical operators in $\mathbf{S}$.

Eq. (6) reveals that a commutator of Lie operators $[\mathcal{F}_1, \mathcal{F}_2]$ is itself a Lie operator associated with a Lie–Poisson bracket of F_1 and F_2,

$$[\mathcal{F}_1, \mathcal{F}_2]I(\boldsymbol{C}) = F_1(\boldsymbol{C})\frac{\partial}{\partial \boldsymbol{C}}\left(F_2(\boldsymbol{C})\frac{\partial}{\partial \boldsymbol{C}}I(\boldsymbol{C})\right) - F_2(\boldsymbol{C})\frac{\partial}{\partial \boldsymbol{C}}\left(F_1(\boldsymbol{C})\frac{\partial}{\partial \boldsymbol{C}}I(\boldsymbol{C})\right), \tag{18}$$

$$= F_1(\boldsymbol{C})\frac{\partial F_2}{\partial \boldsymbol{C}}(\boldsymbol{C}) - F_2(\boldsymbol{C})\frac{\partial F_1}{\partial \boldsymbol{C}}(\boldsymbol{C}), \tag{19}$$

$$= \{F_1(\boldsymbol{C}), F_2(\boldsymbol{C})\}. \tag{20}$$

Expanding the nested commutators that appear in Eq. (16) reveals their form for numerical operators,

$$[\mathcal{F}_1, [\mathcal{F}_2, \mathcal{F}_3]]I(\boldsymbol{C}) = \{F_2(\boldsymbol{C}), F_3(\boldsymbol{C})\}' F_1(\boldsymbol{C}) - F_1(\boldsymbol{C})'\{F_2(\boldsymbol{C}), F_3(\boldsymbol{C})\}, \tag{21}$$

where the primes signify partial differentiation with respect to $\boldsymbol{C}$.

To find the term E_F, replace the two nested commutators in Eq. (16) with their corresponding expressions of Eq. (21). Doing so reveals

$$\frac{\tilde{\partial}}{\partial t}\boldsymbol{C}(t) = \tilde{F}(\boldsymbol{C}) \equiv F_\mathrm{R}(\boldsymbol{C}) + F_\mathrm{D}(\boldsymbol{C}) + \delta t^2\frac{1}{24}\left[F'_\mathrm{DR}(F_\mathrm{R} + 2F_\mathrm{D}) - (F'_\mathrm{R} + 2F'_\mathrm{D})F_\mathrm{DR}\right] + \mathcal{O}(\delta t^4), \tag{22}$$

where

$$F_{\mathrm{DR}} = F'_{\mathrm{R}}F_{\mathrm{D}} - F'_{\mathrm{D}}F_{\mathrm{R}} \tag{23}$$

is the Lie–Poisson bracket from Eq. (18). Eq. (22) contains the original components of Eq. (1) plus terms describing the splitting error and leads to the following expression for E_{F}:

$$E_{\mathrm{F}} = \frac{1}{24}[F'_{\mathrm{DR}}(F_{\mathrm{R}} + 2F_{\mathrm{D}}) - (F'_{\mathrm{R}} + 2F'_{\mathrm{D}})F_{\mathrm{DR}}]. \tag{24}$$

Substituting into this using Eqs. (2), (3), and (23), we arrive at

$$\begin{aligned} E_{\mathrm{F}} = \frac{1}{24}[&(R'(\boldsymbol{C})\boldsymbol{D}\nabla^2(\boldsymbol{C}) - (\boldsymbol{D}\nabla^2(\boldsymbol{C}))'R(\boldsymbol{C}))'(R(\boldsymbol{C}) + 2\boldsymbol{D}\nabla^2(\boldsymbol{C})) - (R(\boldsymbol{C}) + 2\boldsymbol{D}\nabla^2(\boldsymbol{C}))'(R'(\boldsymbol{C})\boldsymbol{D}\nabla^2(\boldsymbol{C}) \\ &- (\boldsymbol{D}\nabla^2(\boldsymbol{C}))'R(\boldsymbol{C}))]. \end{aligned} \tag{25}$$

For all *linear* operators $F_l \in \mathbf{S}$ with *any* operator $F_i \in \mathbf{S}$ the following property holds:

$$F'_l(\boldsymbol{C})F_i(\boldsymbol{C}) \equiv F_l(F_i(\boldsymbol{C})). \tag{26}$$

Applying this to Eq. (25), the term $(\boldsymbol{D}\nabla^2(\boldsymbol{C}))'R(\boldsymbol{C})$ becomes $\boldsymbol{D}\nabla^2(R(\boldsymbol{C}))$. Expanding out the braced expression leads to

$$\begin{aligned} &\boldsymbol{D}R'(\boldsymbol{C})\nabla^2\boldsymbol{C} - \boldsymbol{D}(\nabla^2(\boldsymbol{C}))'R(\boldsymbol{C}) \\ &\quad = \boldsymbol{D}R' \cdot (\boldsymbol{C}_{xx} + \boldsymbol{C}_{yy} + \boldsymbol{C}_{zz}) - \boldsymbol{D}\vec{\nabla} \cdot (\hat{i}(R'\boldsymbol{C}_x + R_x) + \hat{j}(R'\boldsymbol{C}_y + R_y) + \hat{k}(R'\boldsymbol{C}_z + R_z)), \qquad (27) \\ &\quad = \boldsymbol{D}R' \cdot (\boldsymbol{C}_{xx} + \boldsymbol{C}_{yy} + \boldsymbol{C}_{zz}) - \boldsymbol{D}R' \cdot (\boldsymbol{C}_{xx} + \boldsymbol{C}_{yy} + \boldsymbol{C}_{zz}) - \boldsymbol{D}R'' \cdot (\boldsymbol{C}_x\boldsymbol{C}_x + \boldsymbol{C}_y\boldsymbol{C}_y + \boldsymbol{C}_z\boldsymbol{C}_z) \\ &\qquad - 2\boldsymbol{D}(R'_x\boldsymbol{C}_x + R'_y\boldsymbol{C}_y + R'_z\boldsymbol{C}_z) - \boldsymbol{D}(R_{xx} + R_{yy} + R_{zz}), \qquad (28) \end{aligned}$$

where for simplicity, we no longer explicitly write R's dependency on $\boldsymbol{C}$. After canceling terms, Eq. (27) can be incorporated into the expanded expression for Eq. (24).

$$\begin{aligned} E_{\mathrm{F}} = \frac{-\boldsymbol{D}}{24}[&(R'' \cdot (\boldsymbol{C}_x\boldsymbol{C}_x + \boldsymbol{C}_y\boldsymbol{C}_y + \boldsymbol{C}_z\boldsymbol{C}_z) + 2(R'_x\boldsymbol{C}_x + R'_y\boldsymbol{C}_y + R'_z\boldsymbol{C}_z) + (R_{xx} + R_{yy} + R_{zz}))'(R + 2\boldsymbol{D}\nabla^2(\boldsymbol{C})) \\ &- (R + \underline{2\boldsymbol{D}\nabla^2(\boldsymbol{C})})'(R'' \cdot (\boldsymbol{C}_x\boldsymbol{C}_x + \boldsymbol{C}_y\boldsymbol{C}_y + \boldsymbol{C}_z\boldsymbol{C}_z) + 2(R'_x\boldsymbol{C}_x + R'_y\boldsymbol{C}_y + R'_z\boldsymbol{C}_z) + (R_{xx} + R_{yy} + R_{zz}))]. \end{aligned} \tag{29}$$

The terms $\boldsymbol{C}_x$, R_{xx}, etc., represent first and second spatial derivatives taken in the direction of the subscripts. Expanding the first line of Eq. (29) and applying Eq. (26) for the underlined term in Eq. (29) produces

$$\begin{aligned} E_{\mathrm{F}} = \frac{-\boldsymbol{D}}{24}[&(R''' \cdot (\boldsymbol{C}_x\boldsymbol{C}_x + \boldsymbol{C}_y\boldsymbol{C}_y + \boldsymbol{C}_z\boldsymbol{C}_z) + 2(R''_x\boldsymbol{C}_x + R''_y\boldsymbol{C}_y + R''_z\boldsymbol{C}_z) + (R'_{xx} + R'_{yy} + R'_{zz}))(R + 2\boldsymbol{D}\nabla^2\boldsymbol{C}) \\ &- (R' + 2\boldsymbol{D}\nabla^2)(R'' \cdot (\boldsymbol{C}_x\boldsymbol{C}_x + \boldsymbol{C}_y\boldsymbol{C}_y + \boldsymbol{C}_z\boldsymbol{C}_z) + 2(R'_x\boldsymbol{C}_x + R'_y\boldsymbol{C}_y + R'_z\boldsymbol{C}_z) + (R_{xx} + R_{yy} + R_{zz}))]. \end{aligned} \tag{30}$$

Eq. (30) is the general form for E_{F}. For typical simulation geometries, most neighboring grid points will contain the same set of compartments and by extension share the same reaction set. For these homogeneous grid areas, the splitting error can be significantly simplified with spatial derivatives of the reaction operators falling out. Accounting for this yields

$$E_{\mathrm{F}_h} = \frac{-\boldsymbol{D}}{24}[(R''' \cdot (\boldsymbol{C}_x\boldsymbol{C}_x + \boldsymbol{C}_y\boldsymbol{C}_y + \boldsymbol{C}_z\boldsymbol{C}_z))(R + 2\boldsymbol{D}\nabla^2(\boldsymbol{C})) - (R' + 2\boldsymbol{D}\nabla^2)(R'' \cdot (\boldsymbol{C}_x\boldsymbol{C}_x + \boldsymbol{C}_y\boldsymbol{C}_y + \boldsymbol{C}_z\boldsymbol{C}_z))]. \tag{31}$$

Further reduction of the expression can be achieved for most biological systems as reactions are typically first and second order. In such cases, R'' will be a constant and R''' will fall out of the equations entirely, yielding

$$E_{\mathrm{F}_{h,2\mathrm{nd}}} = \frac{\boldsymbol{D}}{24}(R' + 2\boldsymbol{D}\nabla^2)(R'' \cdot (\boldsymbol{C}_x\boldsymbol{C}_x + \boldsymbol{C}_y\boldsymbol{C}_y + \boldsymbol{C}_z\boldsymbol{C}_z)). \tag{32}$$

One approach to the evaluation of Eq. (30) would be to expand out its second line by evaluating ∇^2 across the terms analytically and then numerically calculating the result. It is however more efficient to evaluate the final group of terms at each spatial grid point and subsequently calculate ∇^2 of the values numerically.

As $\delta t^2 E_\mathrm{F}$ has units of μM/s, we take the value of $\delta t^3 E_\mathrm{F}$ as our truncation error estimate in units of concentration. It is this error estimate, along with those of the reaction and diffusion operators that allow for a full estimation of the numerical error for the propagator over a time step δt.

For stiff systems, Press et al. [17] recommend using a relative error above a given threshold concentration value and an absolute error below. We adopt their recommended scaling of $\max(|\boldsymbol{C}|, s)$, and so error values presented in this paper should be interpreted as the maximum *non-negligible* error values used to determine system evolution. Therefore, units for truncation error values will not be specified. The value s is determined by the characteristic scale of the system's concentration values, and we set $s = 1$ for all simulations presented in this paper.

As biochemical reactions are rarely more than bimolecular, Eq. (32) is often the case for systems of equations modeling homogeneous components of biochemical processes. For the purposes of implementation we have kept the generalized form (Eq. (30)) and exploit the sparsity of the high order terms for computational efficiency.

For example, in one dimension the vector term $R''\boldsymbol{C}_x\ \boldsymbol{C}_x$ may be written component-wise as

$$(R''\boldsymbol{C}_x\boldsymbol{C}_x)^{(i)} = \sum_{j,k} \frac{\partial^2 R^{(i)}}{\partial \boldsymbol{C}^{(j)} \partial \boldsymbol{C}^{(k)}} \boldsymbol{C}_x^{(j)} \boldsymbol{C}_x^{(k)}. \tag{33}$$

This term may be evaluated efficiently by only summing over nonzero $\frac{\partial^2 R^{(i)}}{\partial \boldsymbol{C}^{(j)} \partial \boldsymbol{C}^{(k)}}$.

Similarly, one may efficiently evaluate $R'''\boldsymbol{C}_x\boldsymbol{C}_x(R + 2\boldsymbol{D}\nabla^2\boldsymbol{C})$ component-wise using

$$(R'''\boldsymbol{C}_x\boldsymbol{C}_x(R + 2\boldsymbol{D}\nabla^2\boldsymbol{C}))^{(i)} = \sum_{j,k,l} \frac{\partial^3 R^{(i)}}{\partial \boldsymbol{C}^{(j)} \partial \boldsymbol{C}^{(k)} \partial \boldsymbol{C}^{(l)}} \boldsymbol{C}_x^{(j)} \boldsymbol{C}_x^{(k)} (R^{(l)} + 2\boldsymbol{D}\nabla^2 \boldsymbol{C}^{(l)}). \tag{34}$$

This sum is evaluated over terms that are third order or higher, and while it would seem that for a system of 100 reactants Eq. (34) would require a sum over 100^3 iterations, R''' is generally very sparse, if not entirely empty, reducing its cost of calculation considerably.

3. Boundaries and inhomogeneity

In biological cell simulations, the full expression of Eq. (30) is required in regions where cell compartments border or overlap. At these locations, specific elements of R may differ due to the differing sets of reactions that take place in the various compartments. Because we need *partial* derivatives of R, we cannot simply numerically difference values of $R(\boldsymbol{C}(\vec{x}, t), \vec{x})$. Instead we must determine new reaction sets for spatial derivatives of R at these locations.

Consider a three-point quadratic interpolation of R over the evenly spaced grid points x_0, x_1, and x_2. The one-dimensional interpolation is expressed as follows:

$$\begin{aligned} R(\boldsymbol{C}(x), x) &= R(\boldsymbol{C}(x_0), x_0) + (R(\boldsymbol{C}(x_1), x_1) - R(\boldsymbol{C}(x_0), x_0))\left(\frac{x - x_0}{\Delta x}\right) \\ &\quad + \frac{R(\boldsymbol{C}(x_2), x_2) - 2R(\boldsymbol{C}(x_1), x_1) + R(\boldsymbol{C}(x_0), x_0)}{2}\left(\frac{x - x_0}{\Delta x}\right)\left(\frac{x - x_1}{\Delta x}\right). \end{aligned} \tag{35}$$

Taking the first centered partial derivative with respect to x, $\frac{\partial R}{\partial x}\big|_{x_1}$ yields the expression

$$\left.\frac{\partial R}{\partial x}\right|_{x_1} = \frac{R(\boldsymbol{C}(x_1), x_2) - R(\boldsymbol{C}(x_1), x_0)}{2\Delta x}. \tag{36}$$

The second derivative is

$$\left.\frac{\partial^2 R}{\partial x^2}\right|_{x_1} = \frac{R(\boldsymbol{C}(x_1), x_2) - 2R(\boldsymbol{C}(x_1), x_1) + R(\boldsymbol{C}(x_1), x_0)}{\Delta x^2}. \tag{37}$$

The inhomogeneous equations (36) and (37) can be predetermined and then evaluated in the same manner as the standard reaction operator equations.

4. Reaction and diffusion truncation error estimates

In addition to the splitting error calculation, a fully adaptive integration scheme must also be able to determine error estimates for the reaction and diffusion operators. For completeness, we present brief explanations of how we determine truncation error estimates for the two operators as well as present explanations of the methods themselves.

4.1. Evaluation of the error for the diffusion propagator

For propagating the diffusion operator we use an ADI method introduced by Douglas [18,19]. The method easily generalizes to problems of arbitrary dimensionality and is well-suited to our intracellular diffusion problem. In three dimensions it may be written as

$$\frac{\boldsymbol{c}^* - \boldsymbol{c}_n}{\Delta t} = \frac{\alpha}{2}\delta_x^2(\boldsymbol{c}^* - \boldsymbol{c}_n) + \alpha\delta_y^2\boldsymbol{c}_n + \alpha\delta_z^2\boldsymbol{c}_n, \tag{38}$$

$$\frac{\boldsymbol{c}^{**} - \boldsymbol{c}_n}{\Delta t} = \frac{\alpha}{2}\delta_x^2(\boldsymbol{c}^* - \boldsymbol{c}_n) + \frac{\alpha}{2}\delta_y^2(\boldsymbol{c}^{**} - \boldsymbol{c}_n) + \alpha\delta_z^2\boldsymbol{c}_n, \tag{39}$$

$$\frac{\boldsymbol{c}_{n+1} - \boldsymbol{c}_n}{\Delta t} = \frac{\alpha}{2}\delta_x^2(\boldsymbol{c}^* - \boldsymbol{c}_n) + \frac{\alpha}{2}\delta_y^2(\boldsymbol{c}^{**} - \boldsymbol{c}_n) + \frac{\alpha}{2}\delta_z^2(\boldsymbol{c}_{n+1} - \boldsymbol{c}_n), \tag{40}$$

where $\alpha = \frac{D}{\Delta x^2}$ and δ_x^2 is a tridiagonal matrix representing 1D diffusion along a strip of space in the direction of the subscript. The concentration vector $\boldsymbol{c}$ is a vector whose elements are concentrations of a single reactant at successive locations along the grid strip. For a strip in the x direction having periodic boundary conditions, the matrix δ_x^2 is of the form

$$\begin{bmatrix} -2 & 1 & 0 & 0 & \cdots & 0 & 0 & 0 & 1 \\ 1 & -2 & 1 & 0 & \cdots & 0 & 0 & 0 & 0 \\ & & & & \ddots & & & & \\ 0 & 0 & 0 & 0 & \cdots & 0 & 1 & -2 & 1 \\ 1 & 0 & 0 & 0 & \cdots & 0 & 0 & 1 & -2 \end{bmatrix}.$$

Notice that due to periodic boundaries the upper right and lower left corner matrix elements are non-zero, and so will require cyclic tridiagonal solving methods [17].

We can somewhat simplify the above ADI equations by subtracting Eq. (38) from Eq. (39), and Eq. (39) from Eq. (40) and then defining $a = \alpha\delta t$. After doing so, we are left with the following reduced equations:

$$\left(1 - \frac{a}{2}\delta_x^2\right)\boldsymbol{c}^* = \left(1 + \frac{a}{2}\delta_x^2 + a\delta_y^2 + a\delta_z^2\right)\boldsymbol{c}_n, \tag{41}$$

$$\left(1 - \frac{a}{2}\delta_y^2\right)\boldsymbol{c}^{**} = \boldsymbol{c}^* - \frac{a}{2}\delta_y^2\boldsymbol{c}_n, \tag{42}$$

$$\left(1 - \frac{a}{2}\delta_z^2\right)\boldsymbol{c}_{n+1} = \boldsymbol{c}^{**} - \frac{a}{2}\delta_z^2\boldsymbol{c}_n, \tag{43}$$

or in the generalized form for an N dimensional system,

$$\left(1 - \frac{a}{2}\delta_1^2\right)\boldsymbol{c}_1^* = \left(1 + \frac{a}{2}\delta_1^2 + \sum_{i=2}^{N} a\delta_i^2\right)\boldsymbol{c}_n, \tag{44}$$

$$\left(1 - \frac{a}{2}\delta_i^2\right)\boldsymbol{c}_i^* = \boldsymbol{c}_{i-1}^* - \frac{a}{2}\delta_i^2\boldsymbol{c}_n \quad \text{for } i = 2\ldots N, \tag{45}$$

$$\boldsymbol{c}_{n+1} = \boldsymbol{c}_N^*. \tag{46}$$

In one dimension, this is simply the Crank–Nicolson differencing scheme [20]. In three dimensions, determining $\boldsymbol{c}_{n+1}$ consists of evaluating three tridiagonal linear equations, each an $\mathcal{O}(n)$ operation. As the left-hand side matrices $(1-\frac{a}{2}\delta^2)$ possess strict diagonal dominance, they are necessarily non-singular [21,22]. In our own implementation in Cellsim, we make generous use of the Gnu Scientific Libraries for quickly solving these tridiagonal and other linear algebra problems. For both zero flux and periodic boundary conditions, no special treatment is required at the boundaries when calculating the intermediate values, $\boldsymbol{c}^*$ and $\boldsymbol{c}^{**}$.

To estimate the truncation error of the method we first notice that the right side of Eq. (41) contains all the required elements of the first-order FTCS diffusion scheme [17,23], which in three dimensions appears as

$$\hat{\boldsymbol{c}}_{n+1} = (1 + a\delta_x^2 + a\delta_y^2 + a\delta_z^2)\,\boldsymbol{c}_n. \tag{47}$$

Therefore, the second order ADI method contains an embedded first order FTCS method that can be used for a simple truncation error estimate:

$$\Delta_{\mathrm{D}} = \boldsymbol{c}_{n+1} - \hat{\boldsymbol{c}}_{n+1}. \tag{48}$$

The maximum valued element of the vector Δ_{D} is taken as the truncation error estimate for a single time step diffusion propagation.

4.2. Evaluation of the error for the reaction propagator

For propagating chemical kinetics, we employ a Rosenbrock method [17]. Such methods have the general form

$$\boldsymbol{C}_{n+1} = \boldsymbol{C}_n + \sum_{i=1}^{s} b_i\boldsymbol{k}_i, \tag{49}$$

$$\boldsymbol{k}_i = \delta t F_{\mathrm{R}}\left(\boldsymbol{C}_n + \sum_{j=1}^{i-1} a_{ij}\boldsymbol{k}_j\right) + \delta t\boldsymbol{J}\sum_{j=1}^{i}\gamma_{ij}\boldsymbol{k}_j, \tag{50}$$

where a_{ij}, b_i, and γ_{ij} are constants, and $\boldsymbol{J}$ is the Jacobian matrix of the chemical kinetics whose elements $\boldsymbol{J}_{ij} = \frac{\partial F_{\mathrm{R}}(\boldsymbol{C}_i)}{\partial \boldsymbol{C}_j}$. We employ the second order method developed in [24]. The method is written as

$$\begin{aligned} \boldsymbol{C}_{n+1} &= \boldsymbol{C}_n + \frac{3}{2}\delta t\boldsymbol{k}_1 + \frac{1}{2}\delta t\boldsymbol{k}_2, \\ (I-\gamma\delta t\boldsymbol{J})\boldsymbol{k}_1 &= F_{\mathrm{R}}(\boldsymbol{C}_n), \\ (I-\gamma\delta t\boldsymbol{J})\boldsymbol{k}_2 &= F_{\mathrm{R}}(\boldsymbol{C}_n + \delta t\boldsymbol{k}_1) - 2\boldsymbol{k}_1, \end{aligned} \tag{51}$$

where $\gamma_{ij} = \gamma = 1 \pm 1/\sqrt{2}$ is chosen for desired stability properties. Further details can be found in [24]. The method contains an embedded first-order method giving an approximate solution of

$$\hat{\boldsymbol{C}}_{n+1} = \boldsymbol{C}_n + \delta t\boldsymbol{k}_1. \tag{52}$$

Again, subtracting the first-order solution, Eq. (52), from the second order solution, Eq. (51), yields a truncation error estimate of $\Delta_{\mathrm{R}} \equiv \boldsymbol{C}_{n+1} - \hat{\boldsymbol{C}}_{n+1} = \frac{\delta t}{2}\boldsymbol{k}_1 + \frac{\delta t}{2}\boldsymbol{k}_2$. We use the maximum valued element of the vector Δ_{R} as the truncation error estimate for a single time step reaction propagation.

5. The integration scheme

Depending upon the model system being integrated, any of the three sources previously described (reaction, diffusion, and Strang splitting) may contribute to the truncation error. Monitoring all three affords one the most information when adjusting the time step during adaptive integration. If any single source's truncation error is above a maximum tolerance $\Delta_{\max}$ for that source, the error can be reduced to an acceptable value by shrinking the time step value. Correspondingly, if all three truncation errors are below their tolerances, the time step can be increased to improve efficiency for a given desired accuracy. This is the basic approach of our integration scheme.

First, since the splitting calculation is explicit, the truncation error due to splitting can be determined without the need to take the step first. Accordingly our integration scheme checks the splitting error first. If this error surpasses the splitting tolerance, the time step δt is reduced to the maximum value allowed by the tolerance. This can be done without the need to recalculate the splitting error using the smaller step size because the splitting error is directly proportional to δt^3. So, if we calculate a splitting error value of Δ_{fail} which turns out to be larger than the error tolerance $\Delta_{\max}$, we need only reduce the step size by a factor of $(\Delta_{\max}/\Delta_{\text{fail}})^{\frac{1}{3}}$ to find a time step that will result in a splitting error equal to the tolerance. If, on the other hand, the error is less than the tolerance, no adjustment to δt is made.

After determining the splitting error, the system is propagated forward in time an amount $\delta t/2$ by the reaction operator F_{R}. If the step fails the tolerance test, the time step size is reduced and the step is re-taken until the tolerance test is passed. If the step passes the tolerance test, then the value of the next time step ($\delta t'$) is increased and δt is left unchanged.

After this, the system is propagated forward in time an amount δt by the diffusion operator F_{D}. If the step fails the tolerance test, then the time step size is reduced, the partly taken integration step is abandoned, and the entire step is restarted with the reaction propagation. If the diffusion propagation passes

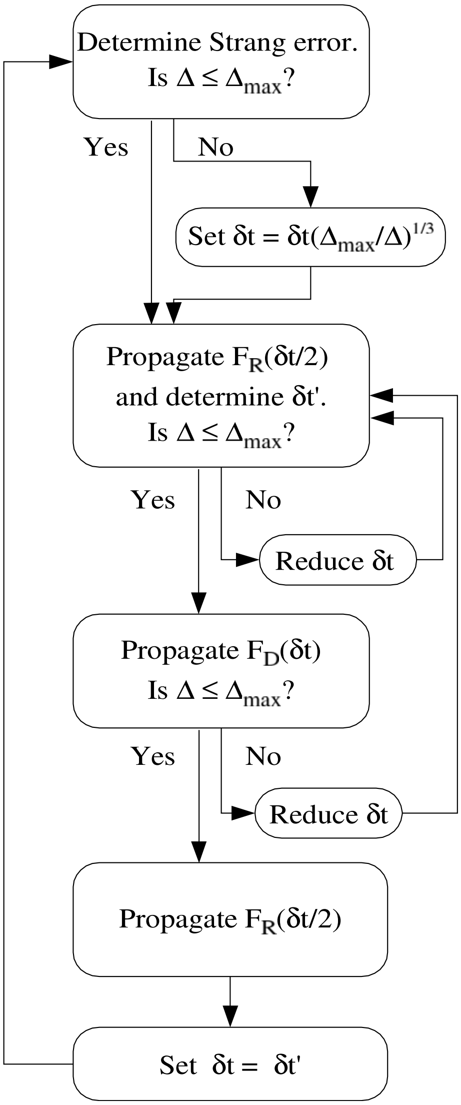

Determine Strang error.
Is $\Delta \leq \Delta_{\max}$?
Yes
No
Set $\delta t = \delta t(\Delta_{\max}/\Delta)^{1/3}$
Propagate $F_R(\delta t/2)$
and determine $\delta t'$.
Is $\Delta \leq \Delta_{\max}$?
Yes
No
Reduce δt
Propagate $F_D(\delta t)$
Is $\Delta \leq \Delta_{\max}$?
Yes
No
Reduce δt
Propagate $F_R(\delta t/2)$
Set $\delta t = \delta t'$

Fig. 1. Algorithm flowchart for a single time step δt. The method consists of four steps: 1. Determine the Strang splitting error and adjust the time step, if needed. 2. Propagate the reaction operator a half time step. 3. Propagate the diffusion operator a full time step. 4. Propagate the reaction operator a half time step. The method checks to see if each error estimate is below the maximum tolerance set for its corresponding propagator. The time step choice δt is considered successful if all error estimates pass this check.

the tolerance test, then the system is propagated forward in time a final half time step $\delta t/2$ by F_{R}. Finally, the value for the next time step is set to the saved value $\delta t'$. A flow chart depicting this algorithm can be found in Fig. 1.

6. Example systems

We now examine three example systems that individually demonstrate system evolution dominated by truncation error from each of the three sources. For the reaction error dominated system, we use a simple auto-catalytic system diffusing through a medium with two slowly diffusing reactants A and B that react quickly to form a dimer. For the diffusion error dominated system, we use a similar system with different time scales for reaction and diffusion constants. For the splitting error dominated system we use a simple simulation of a kinase/phosphatase cascade, activated by a membrane-bound receptor/stimulus complex on a compartmentalized grid. All simulations are integrated using CellSim [5].

As a check for the accuracy of the method, we also integrate out each system using an explicit fourth-order Runge–Kutta (RK4) method and FTCS diffusion with a small fixed time step (10^{-5} s for the two auto-catalytic systems and 10^{-3} s for the kinase cascade system), an accurate but computationally costly method. We plot the maximum concentration difference between the two methods at each time step as cumulative error calculations. Since the time points of the two methods do not identically match, we use simple linear interpolation of the RK4 data to determine appropriate values. We provide these values for the first 100 seconds of each simulation, approaching steady state concentrations in all cases. Computational storage and time constraints prevented us from integrating the RK4 simulations out further.

6.1. Diffusion error dominated system

Diffusion error dominated dynamics are characterized by fast diffusion constants compared to other system rate parameters. In our example of a diffusion error dominated system, we start with a 100×100 square grid of width 2 μm, with an initial Gaussian distribution of the reactants. The relative concentrations of the reactants do not match their kinetic equilibrium concentrations, and the boundary conditions are periodic. The sole kinetic reaction is

$$A + B \rightleftarrows 2A, \tag{53}$$

where the forward rate constant is 0.001 $(\mu\mathrm{M\ s})^{-1}$, the backward rate is 0.0001 $(\mu\mathrm{M\ s})^{-1}$, and the diffusion constant for both reactants is 5.0 $\mu\mathrm{m}^2/\mathrm{s}$. This leads to the reaction and diffusion operators for this system:

$$F_{\mathrm{R}} = \begin{bmatrix} 0.001[A][B] - 0.0001[A][A] \\ 0.0001[A][A] - 0.001[A][B] \end{bmatrix}, \tag{54}$$

$$F_{\mathrm{D}} = 5.0\nabla^2 \begin{bmatrix} [A] \\ [B] \end{bmatrix}. \tag{55}$$

Initial conditions are

$$A(x,y) = \frac{1}{\sqrt{2\pi}\sigma} \mathrm{e}^{-\frac{x^2+y^2}{2\sigma^2}}, \tag{56}$$

$$B(x,y) = 2A(x,y), \tag{57}$$

where $\sigma = 0.205$, and the origin is at the center of the grid. This 2D system is integrated out from time $t_0 = 0$ s to time $t = 1000$ s using the adaptive integration algorithm presented earlier in this paper (Fig. 1). The maximum allowed error for each source of error is $\Delta_{\max} = 0.001$.

The initial dynamics of the reactants can be seen in the first two rows of Fig. 2. Shown are the concentration profiles after the first step ($t_1 = 0.000125$ s), after the initial Gaussians have diffused to approximately half their initial height ($t_2 = 0.00342$ s), and after they have diffused to uniform distributions ($t_3 = 0.0995$ s). From this point the uniformly distributed reactants rise/fall to steady-state reaction values. Note that the legend scales are different for all rows in these image collages.

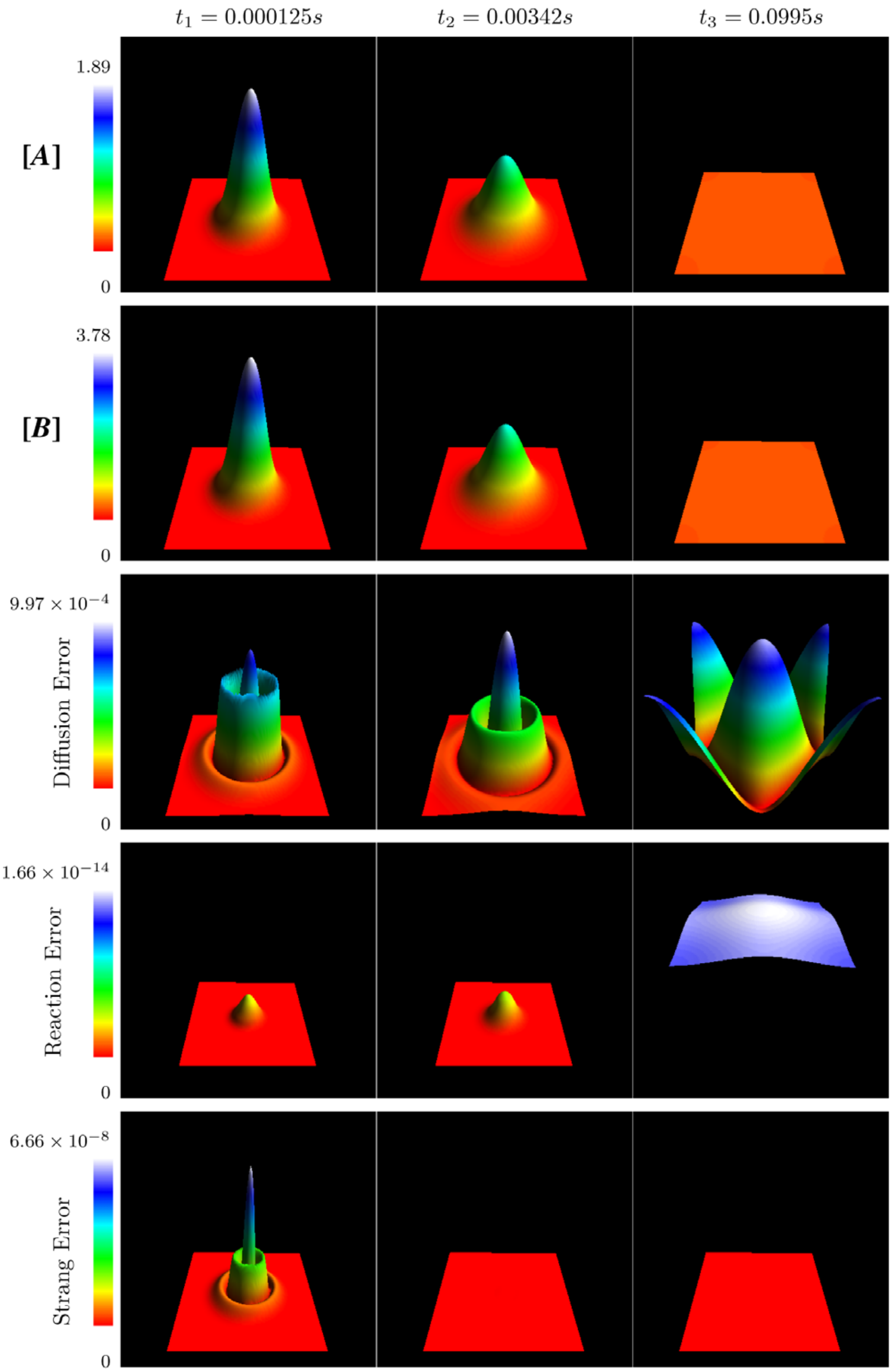


Fig. 2. Detailed concentration and truncation error profiles of the diffusion error dominated system during the initial 0.1 s of simulation. During this initial period the Gaussian distributions of the reactants A and B quickly diffuse out to uniform distributions over the grid. The three columns of images correspond to the simulation times $t_1 = 0.000125$ s, $t_2 = 0.00342$ s and $t_3 = 0.0995$ s. The top two rows show the time evolution of the reactants A and B (in μM concentration), and the bottom three rows show the pointwise maximum truncation error profiles due to diffusion, reaction, and Strang splitting.

The lower three rows of Fig. 2 show the maximum error values over the grids for the three sources of error – diffusion, reaction, and Strang splitting. For time $t_1 = 0.000125$ s, the diffusion error plot contains three points of interest – a central peak corresponding to the central concentration peaks, an inner ring corresponding to the bottom edge of the concentration Gaussian, and a less pronounced third outer ring corresponding to the area just outside the concentration Gaussian. The central peak of truncation error arises simply due to the central curvature of the concentration distribution. The inner and outer truncation error rings correspond to the areas approaching the edge of the concentration Gaussian – from both the zero gradient outside area and the constant gradient slope of the Gaussian. As the concentration Gaussians spread out, these diffusion error rings move radially as well, as shown for time $t_2 = 0.00342$ s. By time $t_3 = 0.0995$ s, each Gaussian has spread out enough to completely fill the grid. The outer error ring disappears, and the inner ring diameter continues to grow larger than the grid width, causing the four peaks in the grid corners, constituting a single peak wrapped around the periodic boundary edge.

At this point, the initial Gaussian concentration distributions have diffused to a uniform distribution over the grid. This lack of strong curvature for concentration causes the diffusion error to drop drastically for small time steps. In Fig. 3, the system diffusion error remains pinned just under the maximum allowed error until around $t = 0.1$ s. At this point the reactant concentrations have become uniform over the grid, which causes the diffusion error to drop several orders of magnitude (from 10^{-3} to 10^{-6}). From this point on, all three sources of error are below the maximum allowed error, and so the adaptive algorithm gradually increases the system time step δt. As the errors do not reach the maximum allowed error until the very end of the run as seen in Fig. 4, δt is increased exponentially from a time step of $\delta t \approx 0.1$ at time $t = 0.1$ s to a time step of $\delta t \approx 500$ s at time $t = 1000$ s (Fig. 4). Only with such large time steps does the diffusion error re-approach the maximum allowed error at this later stage of the simulation.

The reaction error and splitting error do not play a major role in the early stages of the system evolution, and remain several orders of magnitude below the system dominating diffusion error up to time $t = 0.1$ s, as seen in the bottom two rows of Fig. 2 and in Fig. 3. At time $t = 0.1$ s the reaction error does not fall off as does the diffusion and splitting error. This is because the reaction error is most sensitive to kinetic rates and concentration values rather than spatial gradients. The reaction error instead increases exponentially along with δt, though it still remains more than two orders of magnitude smaller than the diffusion error.

The cumulative error for the diffusion dominant system remains low throughout the simulation, ranging between 4.11×10^{-4} μM and 1.31×10^{-5} μM until a uniform reactant distribution is reached around $t = 0.1$ s, at which point the cumulative error drops significantly (Fig. 3).

6.2. Reaction error dominated system

For a reaction error dominated example, we start with the same two Gaussian distributions of A and B reacting to form $2A$ in an autocatalytic reaction. The grid geometry and initial concentration distributions also remain the same. The forward and backward kinetic rate constants are changed to $1\ (\mu\mathrm{M\ s})^{-1}$ and $0.1\ (\mu\mathrm{M\ s})^{-1}$, respectively. The diffusion constant for both reactants is set to 0.001 $\mu\mathrm{m}^2$/s and $\varDelta_{\max} = 0.001$. The reaction and diffusion operators for this system become:

$$F_{\mathrm{R}} = \begin{bmatrix} [A][B] - 0.1[A][A] \\ 0.1[A][A] - [A][B] \end{bmatrix} \tag{58}$$

$$F_{\mathrm{D}} = 0.001\nabla^2 \begin{bmatrix} [A] \\ [B] \end{bmatrix}. \tag{59}$$

The time scale of the reaction operator is now much smaller than the diffusion operator.

In Fig. 5 we examine a series of reactant concentrations and maximum truncation errors over the grid during the initial stages of system evolution – the early reaction error dominated period from time $t = 0$ s to time $t \approx 3$ s. Fig. 6 shows the same concentrations and errors during the later, diffusion error dominated stages of system evolution. As the reaction kinetics reach steady-state values, the reaction error no longer dominates and the system integration evolves at a faster rate that allows the slower diffusion to dominate.

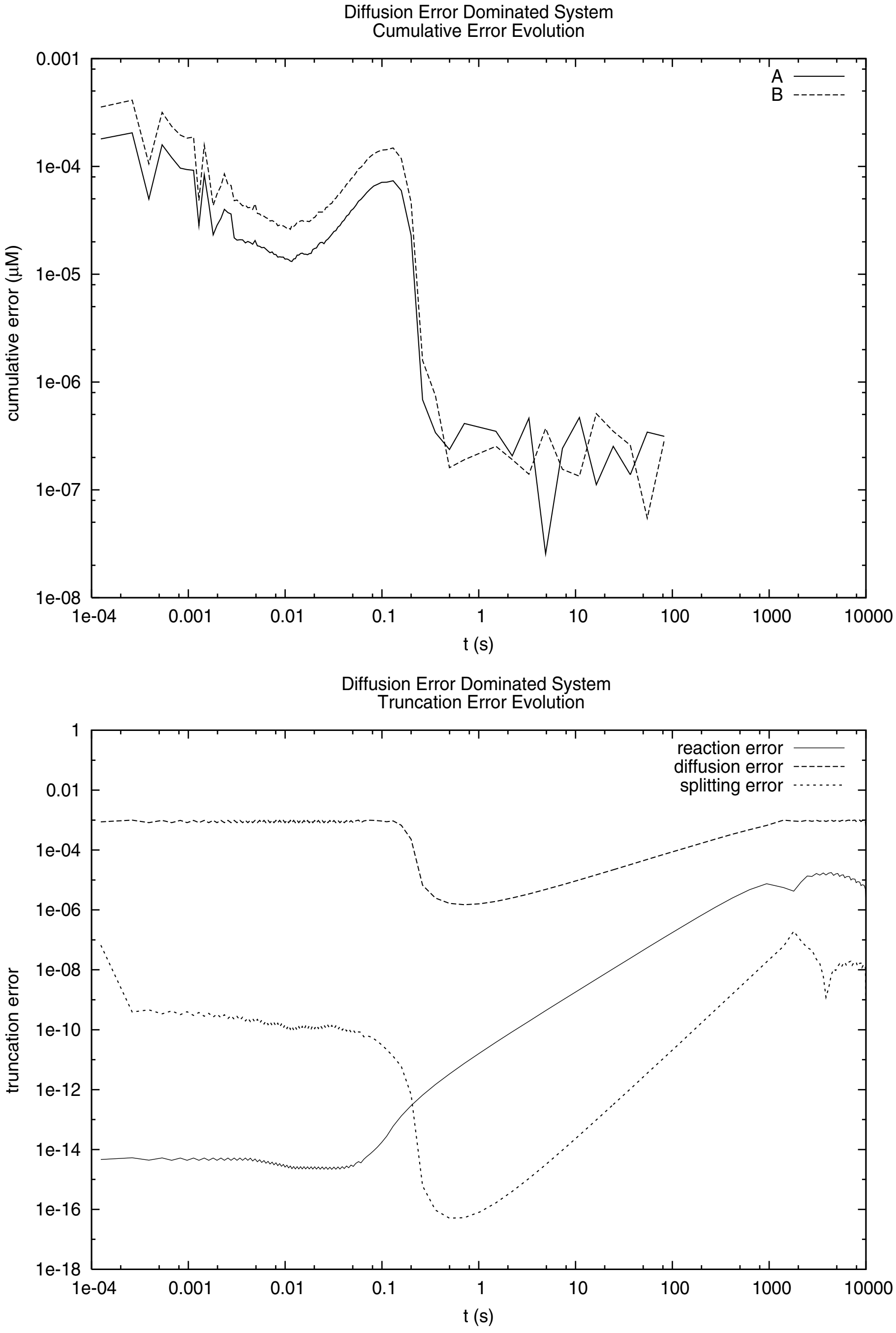


Fig. 3. Maximal truncation error estimates (bottom) and cumulative errors (top) for the diffusion error dominated system. After 0.1 s, the reactants A and B are uniformly distributed over the grid, resulting in reduced truncation error estimates for small time steps. The cumulative error also drops at this point, as the system reaches steady state.

In Fig. 5, we see that the enzymatic reaction kinetics initially dominate, rapidly converting B into A ($t_1 = 0.001$ s to $t_2 = 0.227$ s). This increases the height of A's Gaussian distribution and creates a depression at the center of B's distribution ($t_3 = 0.676$ s). At time $t_4 = 2.776$ s in Fig. 6, the "shoulders" of B's distribution ($t_4 = 2.776$ s) are reduced to reach a final Gaussian distribution at kinetic equilibrium ($t_5 = 15.064$ s).

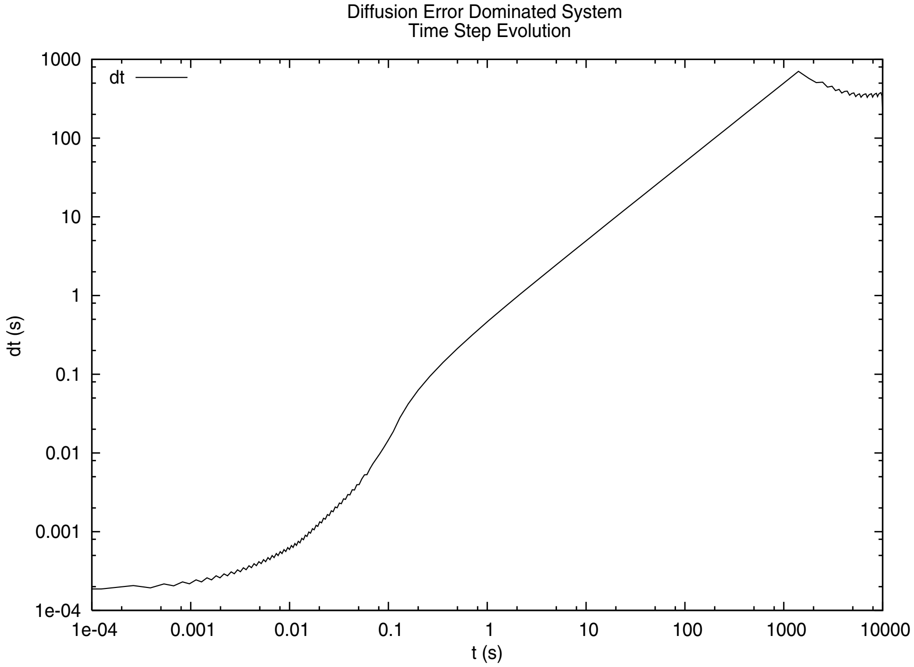


Fig. 4. Evolution of the time step δt for the diffusion error dominated system.

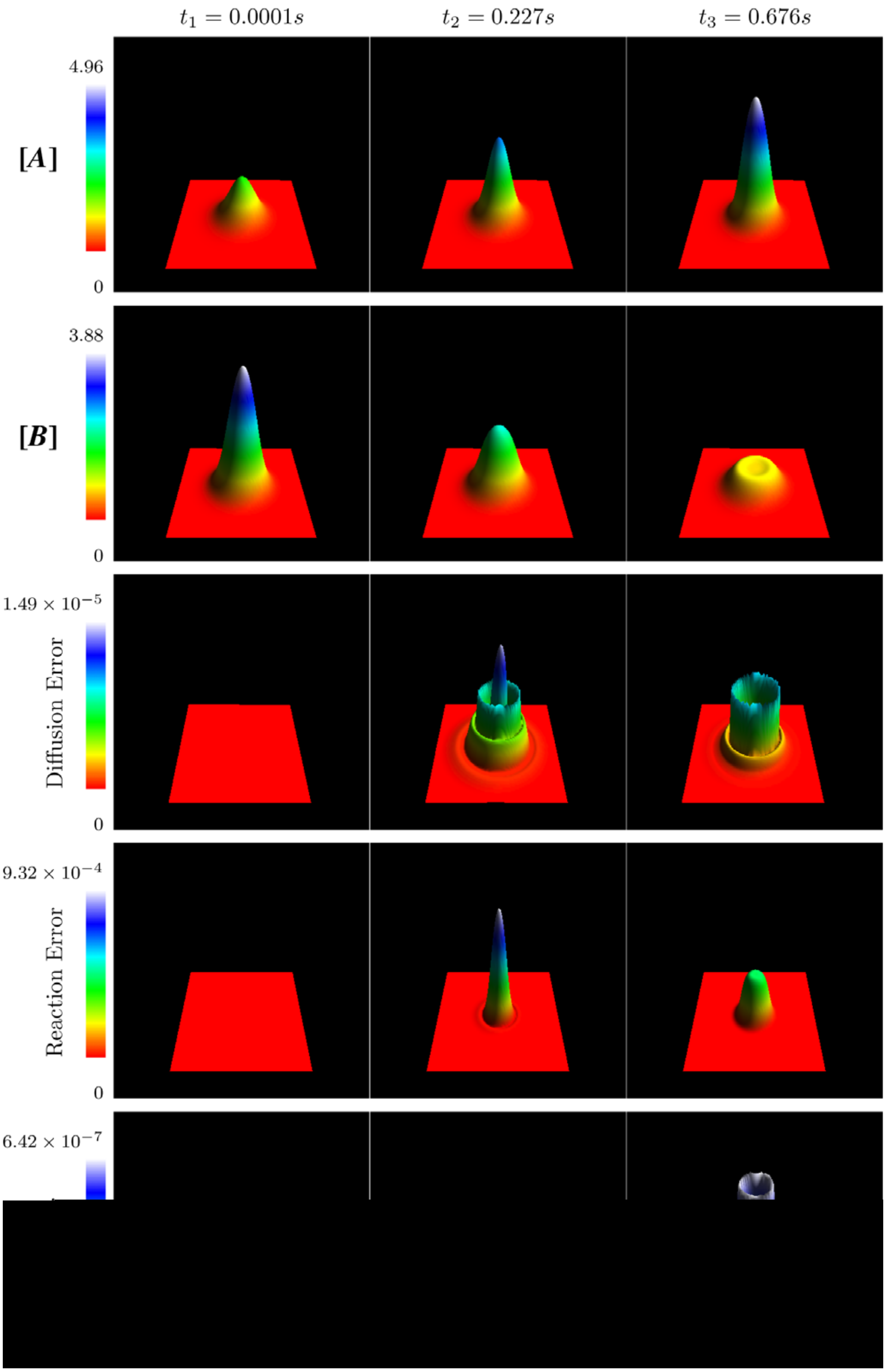
Eventually, near time $t_6 = 918.65$ s, the reactants have diffused uniformly over the grid to steady-state concentrations, which allows for exponential growth of the time step value. As seen in Fig. 8, δt grows over the full extent of the simulation from an initial time step of 10^{-3} s to final steps of over 1000 s.

As for the truncation errors, during the first second of integration the reaction error plays a larger role in the system evolution than the diffusion error and splitting error, though the latter two rise in tandem (Fig. 7). At time $t_2 = 0.227$ s, the reaction error peaks just under the maximum allowed error and then begins decreasing exponentially in three stages: from $t \approx 0.2$ s to $t \approx 1.5$ s after it has been overtaken by the diffusion error, from $t \approx 1.5$ s to $t \approx 100$ s, and from $t \approx 100$ s to $t \approx 200$ s. After the diffusion/reaction crossover around time $t \approx 1.5$ s, the splitting error falls in tandem with the falling reaction error. Beyond $t \approx 200$ s, both reaction and splitting error remain at low values approaching the lower limit of double precision.

During the drop in the reaction error and splitting error, the diffusion error dominates from time $t_4 = 2.776$ s to time $t_6 = 918.65$ s, with the maximum diffusion error hovering just under the maximum allowed error value (Fig. 7). After this point the uniform reactant distributions greatly reduce the diffusion truncation error for small time steps, and so the diffusion error drops two orders of magnitude before the increased time step sizes level the diffusion error magnitude at the end of the simulation. Because of the limited time span of the simulation, truncation error values do not re-approach the maximum allowed value.

Similarities can be seen between the time step evolution and truncation error evolution of the reaction and diffusion error dominated system simulations in their later periods. As seen in Figs. 3 and 7, after the reactants reach uniform distribution, the diffusion error drops by multiple orders of magnitude. This allows δt to grow exponentially until errors re-approach specified limits as seen in Figs. 4 and 8.

The similarities can be further illuminated by comparing Figs. 2 and 6. Differences between the two sets of images can largely be attributed to the ‘‘shoulder’’ of B’s distribution in Fig. 6. For instance, the additional rings present in the diffusion error of the reaction-dominated system (third row in Fig. 6) can be attributed to this. Once the reactants reach kinetic equilibrium around $t \approx 15$ s, the diffusion error evolves quite similarly to that of the diffusion error dominated system, only at a slower time scale. Conversely, the reaction and splitting error profiles evolve differently between the two simulations. In the diffusion error dominated system, the reactants have not reached kinetic equilibrium by the time diffusive equilibrium is reached, so reaction error



[redacted]

Fig. 5. Detailed concentration and truncation error profiles of the reaction error dominated system during the initial 0.676 s of simulation. During this initial period the Gaussian distributions of the reactants A and B react quickly to increase the concentration of A and reduce the concentration of B. The three columns of images correspond to the simulation times $t_1 = 0.0001$ s, $t_2 = 0.227$ s and $t_3 = 0.676$ s. The top two rows show the time evolution of the reactants A and B (in μM concentration), and the bottom three rows show the maximum pointwise truncation error profiles due to diffusion, reaction, and Strang splitting.

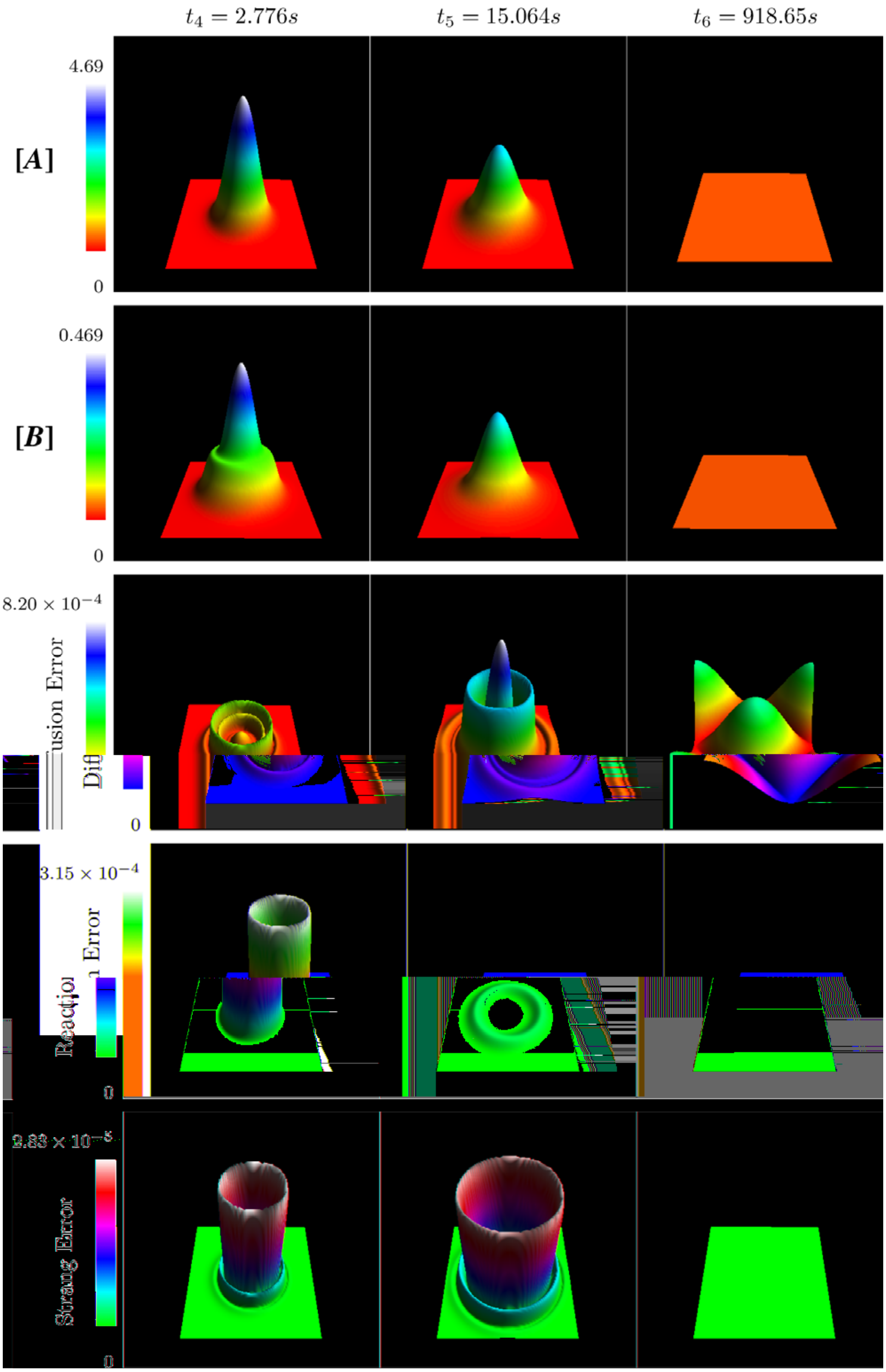


Fig. 6. Detailed concentration and truncation error profiles of the reaction error dominated system during the later, diffusion error dominated stages of simulation. During this period the reactants A and B slowly diffuse to a uniform distribution over the grid. The three columns of images correspond to the simulation times $t_4 = 2.776$ s, $t_5 = 15.064$ s and $t_6 = 918.65$ s. The top two rows show the time evolution of the reactants A and B (in μM concentration), and the bottom three rows show the maximum pointwise truncation error profiles due to diffusion, reaction, and Strang splitting.

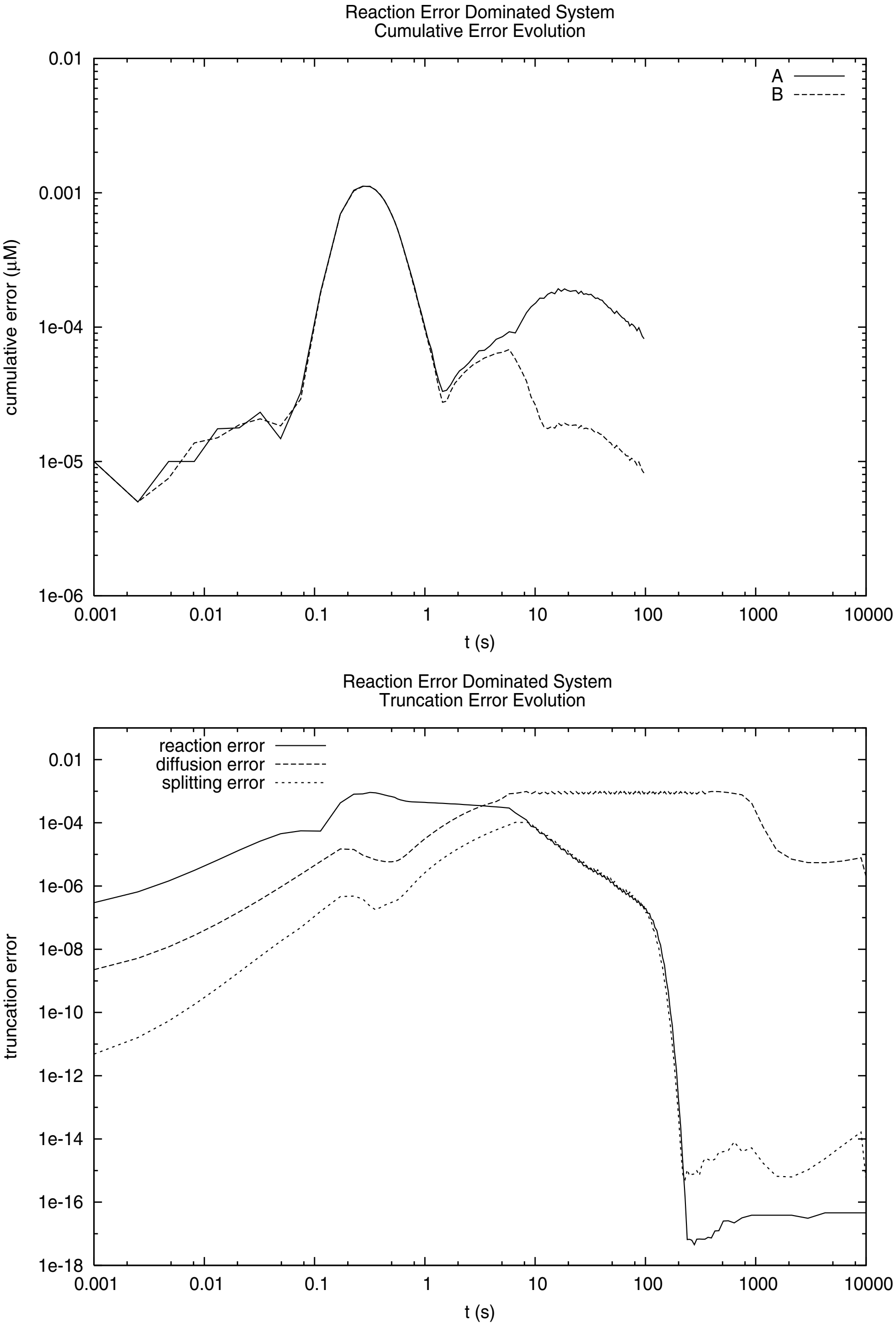


Fig. 7. Truncation error estimates (bottom) and cumulative errors (top) in an initially reaction error dominated system. The reactants *A* and *B* quickly reach kinetic reaction steady-state values at time $t \approx 15$ s, by which point the system has become diffusion error dominated. By time $t = 1000$ s, the reactants have been uniformly distributed over the grid.

values do not fall as they do in the reaction error dominated system. Because the splitting error is affected by both reaction and diffusion terms, it also is dissimilar between the two simulations.

Again, the cumulative error for the reaction dominant system remains low throughout the simulation, ranging between 1.11×10^{-3} μM and 5×10^{-6} μM for the first 100 seconds of simulation (Fig. 7). The initial peak in the cumulative error corresponds to the initial fast reaction-dominated dynamics of the system evolution and is mirrored by a less pronounced peak in the reaction truncation error estimate.

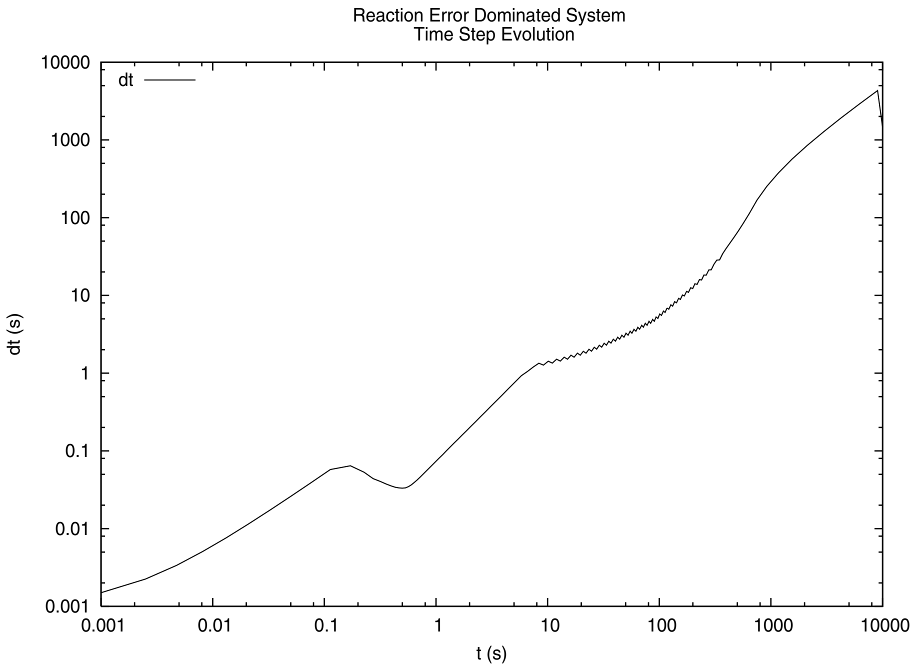


Fig. 8. Evolution of the time step δt for the reaction error dominated system.

6.3. Strang splitting error dominated system

As an example of a system dominated by Strang splitting error, we model a kinase/phosphatase cascade activated by an extracellular stimulus [25]. In our simulation, an extracellular ligand S binds to a transmembrane receptor R, which activates a cytosolic kinase K near the membrane. This activated kinase K^* is free to diffuse within the cytoplasm and undergo dephosphorylation by a phosphatase P. The system equations are as follows:

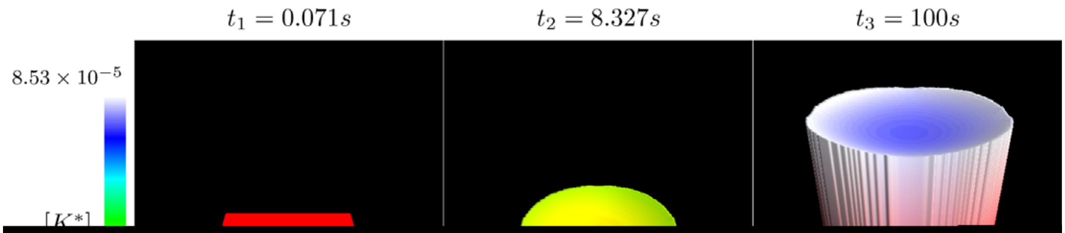
Reaction	k_f	k_b
$S + R \rightleftarrows S.R$	4.2	0.25
$S.R + K \rightleftarrows S.R.K$	1.2	0.8
$S.R.K \rightarrow K^* + S.R$	0.2	
$K^* + P \rightleftarrows K^*.P$	1.98	25
$K^*.P \rightarrow K + P$	6	

with appropriate units of micromolar concentration and seconds. The initial concentrations of the reactants are

S	0.001 μM
R	1.6667 μM
K	0.2 μM
P	0.224 μM

All other concentrations are initially zero, and the stimulus concentration S is held constant. All parameter values are adapted from the MAPK kinase cascade in [26].

The simulation is run on a 100×100 point square grid of total width 5 μm. We model the cell as a circular region whose radius extends to the edge of the grid. The outer grid points of the circular region form the cell



[redacted]

Fig. 9. Activated kinase (K^*) concentration and truncation errors for the Strang splitting error dominated system. The distribution of K^* rises to a steady state by time $t_3 = 100$ s. The splitting error immediately increases during the initial time steps until it peaks at the maximum allowed value of $\Delta_{\max} = 10^{-5}$. The splitting and diffusion errors are most prominent near the membrane region of the cell.

membrane region. The *extracellular* region overlaps with the *membrane* region and extends to the grid edge. The *cytosolic* region also overlaps with the *membrane* region and extend into and fills the cell center. The receptor R and all its complexes exist in and are free to diffuse among grid points containing the membrane compartment. The stimulus S exists in extracellular grid points, and reactants K, K^*, $K^*.P$, and P all exist in the cytosol. The diffusion constant for all reactants is 1 μm^2/s. This system is integrated from time $t = 0$ out to time $t = 200$ s using our adaptive integrator with a tolerance of $\Delta_{\max} = 10^{-5}$ for all three sources of error.

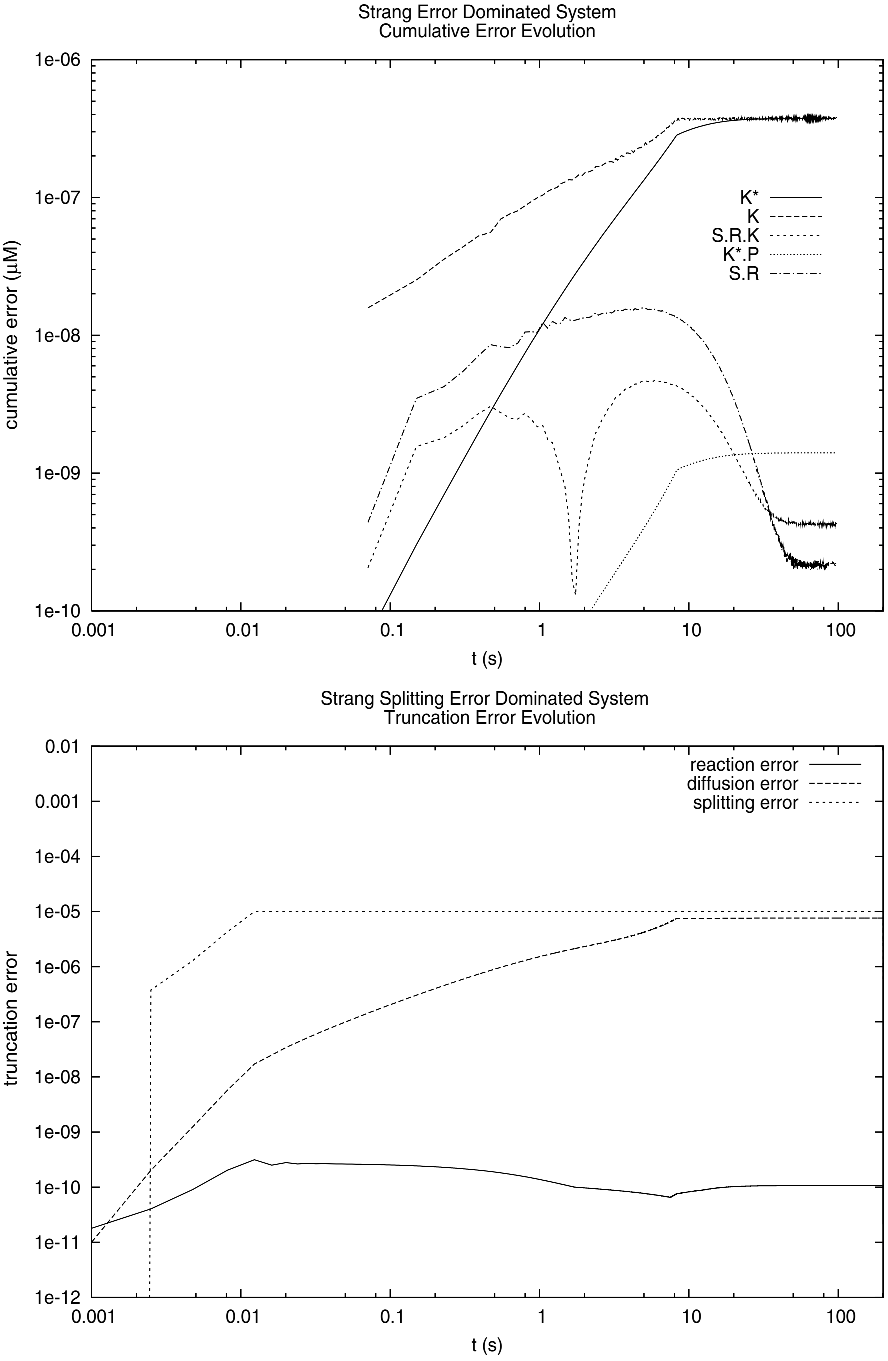


Fig. 10. Truncation error estimates and cumulative errors of the Strang splitting error dominated system. The diffusion error rapidly approaches the splitting error initially, then levels at 0.76 of the maximum allowed value.

We chose this system not only to more directly show the algorithm's applications to problems encountered in systems biology, but also to demonstrate how splitting errors can dominate for compartmentalized, heterogeneous systems of nonlinear reactions. Such systems often occur for spatiotemporal cell signal transduction simulations, where the system reactants (proteins, small molecules, ions, etc.) are kept separate by compartment borders and cell geometry.

As seen in Fig. 9, K^* concentration rises to a steady state of 0.0853 nM at the cell edge which decays exponentially inward to a central concentration of 0.075 nM. Almost immediately ($t_1 = 0.071$ s, first column of Fig. 9), the splitting error dominates the system evolution. The error is most prominent at the interface between the *extracellular* and *membrane* compartments located at the outer edges of the cell. After sufficient accumulation of $S.R$ in the membrane, splitting error increases at the interface between the *membrane* and *cytosolic* compartments located at the internal edges of the cell, eventually overtaking the outer-edge splitting error at time $t_2 = 8.327$ s (Fig. 9, second column). The splitting error at the outer edge reduces slightly by time $t_3 = 100$ s and the system reaches steady state (Fig. 9, third column).

In Fig. 9, the reaction error (third row) quickly reaches a peak value at t_1, which then reduces to the distributions seen at times t_2 and t_3. The diffusion error distribution (second row) mimics that of the splitting error and gradually increases in value up to a maximum of 7.63×10^{-6} at time t_2, just below the allowed tolerance.

As seen in Fig. 10, for the vast majority of integration time, the splitting error is at the maximum allowed value. At steady state the diffusion error is three-quarters of this value and the reaction error is 5 orders of magnitude smaller. Therefore, the splitting error determines the time step evolution plotted in Fig. 11. During the initial integration steps, the time step size increases exponentially until the splitting error reaches the maximum allowed value at time $t \approx 0.03$. From here the time step size increases more slowly until time $t_2 = 8.327$ s. From this time on through the remainder of the simulation, the time step size does not significantly change.

Of the seven reactants present in the simulation, it is inactive K that produces the greatest splitting error throughout the simulation. This is localized at the cell membrane where K partakes in the most reactions, involving $S.R$, $S.R.K$, and $K^*.P$ simultaneously. It is interesting to note that the splitting error for the reactants S, R, $S.R$ and $S.R.K$ quickly drop off to negligible values after time $t_2 = 8.327$ s.

The discontinuous appearance of the splitting and diffusion error is a result of modeling a circular cell on a square grid lattice. Membrane grid points will have neighboring grid points of varying compartments, depending on the circular membrane's orientation to the square lattice direction. Therefore, the three-point numerical differencing used in the diffusion error and splitting error calculations will be affected by the location on the membrane. Despite the random appearance, the concentration and error profiles all have four-fold symmetry (eight-fold mirror symmetry), as would be expected on a square grid.

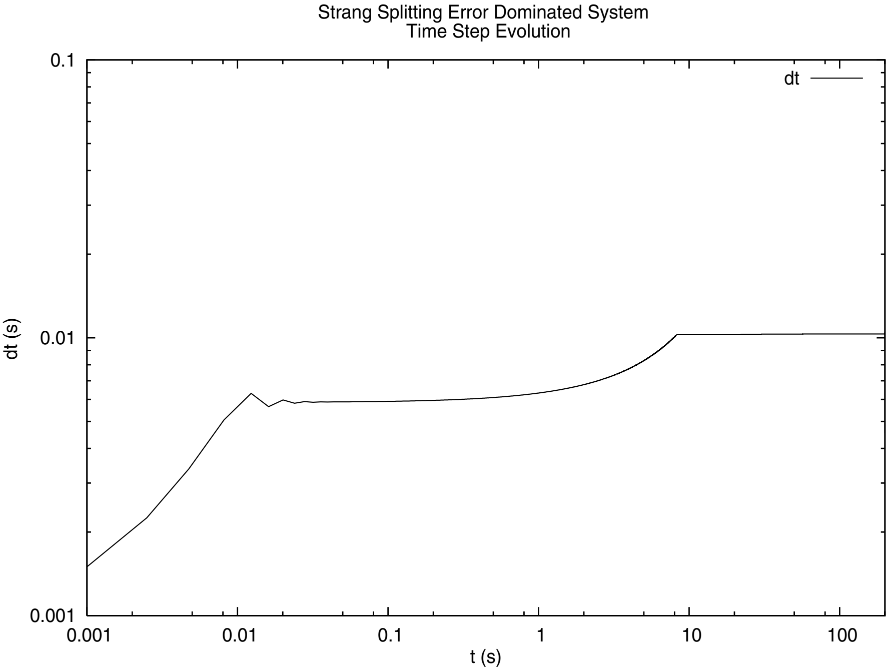


Fig. 11. Evolution of the time step δt for the Strang splitting error dominated system. The integration time step reaches a steady-state value of 0.01026 s at time $t_2 = 8.327$ s.

The cumulative error for the Strang splitting dominant system is highest for the kinase K, as the total error for both the active and inactive form reach values of 3.75×10^{-7} μM within the first 12 seconds of the simulation (Fig. 10).

7. Discussion and final remarks

We have shown that the evolution of even the simplest reaction–diffusion systems likely to be encountered in multi-dimensional systems biology simulations can be dominated by truncation error due to both reaction and diffusion operators, as well as the truncation error due to the common second-order Strang's method of operator splitting. In multi-dimensional cell signaling simulations, truncation error due to Strang splitting can entirely dominate the system evolution as demonstrated by our third example simulation.

We have presented a method of monitoring the truncation error due to Strang splitting and incorporating it into an adaptive step size integration algorithm. While truncation error and global error monitoring and control of individual operators is a common field of study [27,28], time adaptive methods which incorporate Strang splitting truncation error as an input to time step control have, to our knowledge, not been previously evaluated.

Operator splitting and time splitting methods have found popularity in a wide variety of applications including Bose–Einstein condensation [29], quantum statistical calculations [30], optical interactions in media [31], and transport in porous media [32]. Moreover, Strang splitting is used extensively in the field of atmospheric simulation. This lead Lanser and Verwer [1] to conjecture that splitting errors are kept within practical bounds for typical problems encountered in the field. As we have shown, this statement cannot be made, in general, for problems in multi-dimensional cell signal modeling. We suggest that the inherent compartmental nature of cells is a significant contributor to the splitting error in modeling cells.

Unlike atmospheric problems, where the chemistry is reasonably smooth spatially (it is a function of temperature, altitude, pressure, humidity, etc., all of which are continuous variables), cells contain clearly defined compartments and borders, over which chemistry can discontinuously vary. While more complex models of cell signaling may take into account such factors as temperature and pressure, which are indeed important for a multitude of cell functionality, the inclusion of locality into even the simplest models of cell signaling requires the presence of compartments and physical borders. Such structures create abrupt sources and sinks within the model chemistry which substantially contribute to the splitting error.

As shown by Lanser and Verwer [1], a complete and general splitting error calculation for reaction–advection–diffusion problems is too cumbersome to be of much practical use. As such, they prescribe a number of simplifications to make the calculation more manageable. Since we are not currently modeling advection, the splitting expression is simplified considerably. While this simplification makes the calculation more feasible, it is still at a significant time cost per step. In our simulations, we have only seen significant calculation time advantages for problems where uniform distributions of reactants reduce the role played by diffusion. In such situations the time step size can increase dramatically (as seen in the reaction error dominated and diffusion error dominated examples), and significantly reduce the calculation time. For more common heterogeneous problems, calculating the various truncation errors every N steps reduces the calculation cost of each by a factor of N, although this makes the method's time adaptivity less robust. The time cost to calculate reaction and diffusion errors using the methods presented in this paper are near negligible.

Currently the most common approach for modeling intracellular signaling networks is based on purely kinetic reaction systems modeled using ODEs, such as Gepasi [33] and Genesis [34]. This is referred to as the "well mixed model" as it assumes homogenous chemistry throughout the cell. Focus has also recently extended to modeling transport phenomena requiring the use of spatially resolved stochastic approaches such as MCell [35] and StochSim [36] as well as spatially resolved kinetic approaches like Virtual Cell [37] and Cell-Sim [5]. The last of these approaches, or possibly even a hybrid of the last two methods, is the most suited for utilization of the Strang splitting error monitoring method outlined in this paper.

Other adaptive methods for solving partial differential equations include adaptive mesh methods [38] and multigrid methods [39], time adaptive splitting methods for quickly equilibrating diffusion [40], and *in situ* adaptive tabulation [13], essentially a storage and retrieval system. The main benefit of our method is ease of implementation into software packages that use fixed Cartesian grids. The method is independent of the underlying reaction and diffusion integrators, and for already existing implementations of Strang splitting only the splitting error calculation and the simple adaptive control method need to be added. However, we have recently proposed methods to extend splitting error control to adaptive mesh refinement techniques, providing adaptivity in space as well as time.

As a possible extension to this work, there are important intracellular advective processes such as active transport along actin filaments which are interesting in that they can be entirely decoupled from diffusion [41]. In other words, reactants undergoing such advection are not simultaneously diffusing. Adding an advection propagator to the Strang split calculation for such reactants introduces splitting error due to the coupling of reaction and advection, while for such reactants diffusion no longer contributes to the splitting error. As such, determining the significance of advection–reaction splitting error in reaction–diffusion/advection cell models is one possible step for further investigation into the role of splitting error in spatiotemporal cell modeling.

Acknowledgments

We thank Rui Zou for help with preparing the manuscript. This work was supported in part by a NSF Bioinformatics Starter Grant and by a grant from the Pennsylvania Tobacco Health Funds Initiative.

References

[1] D. Lanser, J.G. Verwer, Analysis of operator splitting for advection–diffusion–reaction problems from air pollution modelling, J. Comput. Appl. Math. 111 (1999) 201–216.

[2] G. Strang, On the construction and comparison of difference schemes, SIAM J. Numer. Anal. 5 (1968) 506–517.

[3] R.I. McLachlan, On the numerical integration of ordinary differential equations by symmetric composition methods, SIAM J. Sci. Comput. 16 (1995) 151–168.

[4] R.I. McLachlan, G.R.W. Quispel, Splitting methods, Acta Numer. 11 (2002) 341–434.

[5] A. Ghosh, D. Miller, R. Zou, H. Sokhansanj, A. Kriete, Computational Systems Biology, Academic Press, Burlington, 2005 (Ch. Spatio-Temporal Systems Biology, pp. 327–362).

[6] I. Free Software Foundation, http://www.gnu.org/copyleft/gpl.html, GNU Public Licence (1989, 1991).

[7] R. Gilmore, Lie Groups, Lie Algebras, and Some of Their Applications, Krieger Publishing Company, Malabar, 1974.

[8] J.M. Sanz-Serna, The State of the Art in Numerical Analysis, Clarendon Press, Oxford, 1997 (Ch. Geometric Integration, pp. 121–143).

[9] J.M. Sanz-Serna, M.P. Calvo, Numerical Hamiltonian Problems, Chapman & Hall, London, 1994.

[10] B. Sportisse, An analysis of operator splitting techniques in the stiff case, J. Comput. Phys. 161 (2000) 140–168.

[11] R. Kozlov, A. Kvearno, B. Owren, The behavior of the local error in splitting methods applied to stiff problems, J. Comput. Phys. 195 (2004) 576–593.

[12] S. Descombes, M. Massot, Operator splitting for nonlinear reaction–diffusion systems with an entropic structure: singular perturbation and order reduction, Numer. Math. 97 (2004) 667–698.

[13] M.A. Singer, S.B. Pope, Exploiting isat to solve the reaction–diffusion equation, Combust. Theory Modelling 8 (2004) 361–384.

[14] J.E. Campbell, On the theory of continuous transformation groups, Proc. Lond. Math. Soc. 28 (1897) 381–390.

[15] E.B. Dynkin, On the representation of the series $\log(e^x e^y)$ for non-commutative x and y by commutators, Math. Sbornik 25 (1949) 155–162.

[29] W. Bao, J. Shen, A fourth-order time-splitting Laguarre–Hermite pseudospectral method for Bose–Einstein condensates, SIAM J. Sci. Comp. 26 (2005) 2010–2028.

[30] S. Chin, Quantum statistical calculations and symplectic corrector algorithms, Phys. Rev. E 69 (2004) 046118.

[31] W. Bao, C. Zheng, A time-splitting spectral method for three-wave interactions in media with competing quadratic and cubic nonlinearities, Commun. Comput. Phys. 2 (2007) 123–140.

[32] K.H. Karlsen, K.-A. Lie, J.R. Natvig, H.F. Nordhaug, H.K. Dahle, Operator splitting methods for systems of convection–diffusion equations: nonlinear error mechanisms and correction strategies, J. Comput. Phys. 173 (2001) 636–663.

[33] P. Mendes, Biochemistry by numbers: simulation of biochemical pathways with Gepasi 3, Trends Biochem. Sci. 22 (1997) 361–363.

[34] J. Bower, D. Beeman, The Book of GENESIS: Exploring realistic Neural Models with the General Neural Simulation System, Springer, New York, 1994.

[35] T.M.B. Joel, R. Stiles, Computational Neuroscience: Realistic Modeling for Experimentalists, CRC Press, 2001 (Ch. Monte Carlo Methods for Simulating Realistic Synaptic Microphysiology Using MCell, pp. 87–127).

[36] N.L. Novre, T.S. Shimizu, Stochsim: modelling of stochastic biomolecular processes, Bioinformatics 17 (2001) 575–576.

[37] L.M. Loew, J.C. Schaff, The Virtual Cell: a software environment for computational cell biology, Trends Biotechnol. 19 (2001) 401–406.

[38] P.A. Zegeling, H.P. Kok, Adaptive moving mesh computations for reaction–diffusion systems, J. Comput. Appl. Math. 168 (1–2) (2004) 519–528.

[39] C.-S. Chou, Y.-T. Zhang, R. Zhao, Q. Nie, Numerical methods for stiff reaction–diffusion systems, Discr. Cont. Dyn. Syst. B 7 (2007) 515–525.

[40] J.C. Schaff, B.M. Slepchenko, Y.-S. Choi, J. Wagner, D. Resasco, L.M. Loew, Analysis of nonlinear dynamics on arbitrary geometries with the Virtual Cell, Chaos 11 (2001) 115–131.

[41] A.-T. Dinh, T. Theofanous, S. Mitragotri, A model for intracellular trafficking of adenoviral vectors, Biophys. J. 89 (2005) 1574–1588.