Algebraic stabilization of explicit numerical integration for extremely stiff reaction networks

Mike Guidry *

Department of Physics and Astronomy, University of Tennessee, Knoxville, TN 37996-1200, USA
Physics Division, Oak Ridge National Laboratory, Oak Ridge, TN 37830, USA
Computer Science and Mathematics Division, Oak Ridge National Laboratory, Oak Ridge, TN 37830, USA

ABSTRACT

In contrast to the prevailing view in the literature, it is shown that even extremely stiff sets of ordinary differential equations may be solved efficiently by explicit methods if limiting algebraic solutions are used to stabilize the numerical integration. The stabilizing algebra differs essentially for systems well-removed from equilibrium and those near equilibrium. Explicit asymptotic and quasi-steady-state methods that are appropriate when the system is only weakly equilibrated are examined first. These methods are then extended to the case of close approach to equilibrium through a new implementation of partial equilibrium approximations. Using stringent tests with astrophysical thermonuclear networks, evidence is provided that these methods can deal with the stiffest networks, even in the approach to equilibrium, with accuracy and integration timestepping comparable to that of implicit methods. Because explicit methods can execute a timestep faster and scale more favorably with network size than implicit algorithms, our results suggest that algebraically-stabilized explicit methods might enable integration of larger reaction networks coupled to fluid dynamics than has been feasible previously for a variety of disciplines.

© 2012 Elsevier Inc. All rights reserved.

1. Introduction

Problems from many disciplines require solving large coupled reaction networks. Representative examples include reaction networks in combustion chemistry [1], geochemical cycling of elements [2], and thermonuclear reaction networks in astrophysics [3,4]. The differential equations used to model these networks usually exhibit stiffness, which arises from multiple timescales in the problem that differ by many orders of magnitude [1,5–7]. Sufficiently complex physical systems often involve important processes operating on widely-separated timescales, so realistic problems tend to be at least moderately stiff. Some, such as astrophysical thermonuclear networks, are extremely stiff, with 10–20 orders of magnitude between the fastest and slowest timescales in the problem. Our concern here is with stiffness as a numerical issue, but we remark that stiffness can have important physical implications because complex processes often function as they do precisely because of the coupling of very slow and very fast scales within the same system.

Books on numerical and computational methods routinely state [1,6,7] that stiff systems cannot be integrated efficiently using explicit finite-difference methods because of stability issues: for an explicit algorithm, the maximum stable timestep is set by the fastest timescales, even if those timescales are peripheral to the main phenomena of interest. The standard

---

 * Address: Department of Physics and Astronomy, University of Tennessee, Knoxville, TN 37996-1200, USA.
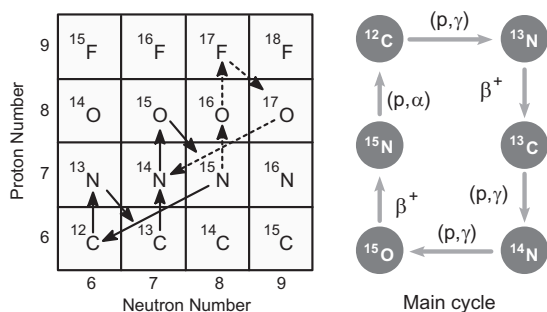   *E-mail address:* guidry@utk.edu

Fig. 1. The CNO (carbon–nitrogen–oxygen) cycle. On the left side the main branch of the cycle is illustrated with solid arrows and a side branch is illustrated with dashed arrows. On the right side, the main branch of the CNO cycle is illustrated in more detail.

resolution of the stiffness problem uses implicit integration, which is stable for stiff systems but entails substantial computational overhead because it requires the inversion of matrices at each integration step. Because of the matrix inversions, implicit algorithms tend to scale from quadratically to cubically with network size unless favorable matrix structure can be exploited. Thus, implicit methods can be expensive for large networks.

A simple but instructive example of stiffness is provided by the CNO cycle for conversion of hydrogen to helium, which powers main-sequence stars more massive than the Sun (Fig. 1). In the CNO cycle the fastest rates under characteristic stellar conditions are β-decays with half-lives $\sim$100 s, but to track the complete evolution of main-sequence hydrogen burning may require integration of the network for hydrogen burning over timescales as large as billions of years ($\sim 10^{16}$ s). If one tries to implement this integration using explicit forward differencing, the largest stable integration timestep will be set by the fastest rates and will be of order 10^2 s. Thus 10^{14} or more explicit integration steps could be required to integrate the CNO cycle to hydrogen depletion. Conversely, typical implicit integration schemes can take stable and accurate timesteps equal to 1–10% of the local time over most of the integration range, and would compute the above numerical integration in a few hundred implicit steps. By virtue of examples such as this, it is broadly accepted that explicit methods are not viable for stiff networks. To quote the authoritative reference *Numerical Recipes* [7], "For stiff problems we *must* use an implicit method if we want to avoid having tiny stepsizes."

Our main interest lies in simulations where the reaction network is one part of a broader problem. Let us take as representative astrophysical thermonuclear networks coupled to multidimensional hydrodynamics. The hydrodynamical evolution controls network conditions (temperature, density,...), and the network influences the hydrodynamics through energy production and modification of composition. Solution of large networks by the usual means is costly in this context and even ambitious simulations use only small networks, or replace the network entirely by parameterization. Then a more realistic network is used in a separate "post-processing" step, where fixed hydrodynamical profiles computed in the original simulation specify the variation of temperature and density with time. Such approximations are especially at issue for problems like Type Ia supernovae, which are 3D asymmetric explosions powered by a complex reaction network releasing energy greater than that of a large galaxy on a timescale of order 1 s.

Astrophysical reaction networks have been used to illustrate, but problems in various fields exhibit a similar complexity. For example, in astrochemical kinetics large chemical evolution networks must be modeled in dynamical environments such as contracting molecular clouds, or in combustion chemistry the burning networks are strongly coupled to dynamical simulations of the air and fuel mixture. Realistic networks in all such applications may be quite large. Modeling combustion of larger hydrocarbon molecules or soot formation can require hundreds to thousands of reacting species with up to 10,000 reactions [1], and realistic networks for supernova explosions imply hundreds to thousands of nuclear isotopes with tens of thousands of reaction couplings [3]. In all such cases current techniques do not permit the coupling of realistic reaction networks to the full dynamics and highly-schematic reaction networks are used in even the most realistic contemporary simulations.

2. Stiffness under equilibrium and non-equilibrium conditions

The pessimism engendered by the Introduction notwithstanding, it would be highly desirable to integrate large, complex networks by explicit means because explicit algorithms are simple and economical, and scale favorably with network size. In principle, this could be accomplished by identifying conditions under which some equations in the network have an approximate analytical solution and using that information to remove algebraically the stiffest components from the general numerical solution, thereby replacing the original network with an approximation that permits larger stable explicit-integration timesteps. To that end, the first task is to understand clearly the nature of the stiffness that we wish to remove from the equations.

2.1. Varieties of stiffness

There are several fundamentally different sources of stiffness instability in reaction networks that are often not clearly distinguished in the literature. Pedagogically, discussions often emphasize an instability associated with small populations becoming negative because of an overly-ambitious numerical integration step, thus converting stable decaying-exponential terms into unstable growing exponentials. However, there are other instabilities that can occur, even when no population variables become negative in an integration step. In these types of instabilities one ends up having to take the difference of large numbers to obtain a result very near zero. The numerical errors that ensue in a standard explicit approach can then accumulate rapidly and destabilize the network, even before any populations become negative. This is still a stiffness instability because the problem results ultimately from a numerical integration attempting to cope with highly-disparate timescales, but the origin of these timescales is different from that discussed above. In this case the disparate timescales are the fast reactions driving the system to equilibrium contrasted with the slow timescale associated with equilibrium itself (which tends to infinity). To illustrate, consider the form of the equations that we desire to solve:

$$\frac{dy_i}{dt} = F_i^+ - F_i^- = (f_1^+ + f_2^+ + \cdots)_i - (f_1^- + f_2^- + \cdots)_i = (f_1^+ - f_1^-)_i + (f_2^+ - f_2^-)_i + \cdots = \sum_j (f_j^+ - f_j^-)_i, \tag{1}$$

where the y_i ($i = 1, \ldots, N$) describe the dependent variables (species abundances for our examples), t is the independent variable (time in our examples), the fluxes between species i and j are denoted by $(f_j^\pm)_i$, and the sum for each variable i is over all variables j coupled to i by a non-zero flux $(f_j^\pm)_i$. For an N-species network there will be N such equations in the populations y_i, with these equations generally coupled to each other because of the dependence of the fluxes on the different y_j.

In Eq. (1) several different ways to group the terms on the right side have been indicated, with the first form representing a decomposition into total flux in and out of species i and the last form separating the total flux into in and out contributions from individual reactions. The utility of these alternative decompositions will be elaborated further below.

2.2. The approach to equilibrium

In this discussion we shall employ the term "equilibration" broadly to mean that evolution of the network is being influenced strongly by nearly-canceling terms on the right sides of the differential equations (1). Two qualitatively different equilibrium conditions may be distinguished that require different algebraic approaches to stabilization.

2.2.1. Macroscopic equilibration

One class of equilibrium conditions results if $F_i^+ - F_i^- \to 0$ (asymptotic) or $F_i^+ - F_i^- \to$ constant (steady-state). Let us refer to these conditions as a *macroscopic equilibration*, since they are statements about entire differential equations in Eq. (1). We shall introduce asymptotic and quasi-steady-state approximations exploiting these conditions that remove whole differential equations from the numerical integration for a network timestep, replacing them with algebraic solutions. Such approximations still integrate the full original set of equations, but they reduce the number of equations integrated *numerically by forward difference*. This helps with stiffness because integration of some or all of the stiffest equations is replaced by a stable analytical solution, and any equations that remain to be integrated numerically tend to have smaller disparities in timescales and thus less stiffness.

2.2.2. Microscopic equilibration

In Eq. (1), F_i^+ and F_i^- for a species i each consist of various terms depending on the other populations in the network. Groups of individual terms on the right side of Eq. (1) may come into equilibrium (so that the sum of their fluxes tends to zero), even if the macroscopic conditions for equilibrium for the entire equation are not satisfied. We shall term this *microscopic equilibration*. Then we may consider an algebraic approximation that removes groups of such terms from the numerical integration by replacing their sum of fluxes with zero. This does not reduce the number of equations integrated numerically, but can reduce their stiffness by removing terms with fast rates, thereby reducing the disparity between the fastest and slowest timescales in the system. Such considerations will be the basis of the partial equilibrium methods to be discussed in Section 5.

3. Reaction networks in the context of larger systems

Assume the coupling of reaction networks to a larger system to be done using operator splitting, where the larger system (the hydrodynamical solver in our examples) is evolved for a timestep holding the parameters computed from the network in the previous step constant, and then the network is evolved over the time corresponding to the hydrodynamical timestep while holding the new variables calculated in the hydrodynamical timestep constant. This places two strong constraints on methods:

(1) The network must be advanced with new initial conditions for each hydrodynamical step. Thus, algorithms must initialize simply and quickly.

(2) Integration of the network must not require time substantially larger than that for the corresponding hydrodynamical timestep.

Let us elaborate further on this second point. Taking the multidimensional, adaptive-mesh, explicit hydrodynamical code Flash [8] applied to Type Ia supernova explosions as representative, we estimate in Ref. [9] that a simulation with realistic networks can be done in a tractable amount of time if the algorithm can take timesteps Δt over the full hydrodynamical integration range that average at least $(0.01 - 0.001) \times t$, where t is the elapsed time in the integration. Timesteps of this size are possible with implicit and semi-implicit algorithms, but those methods are inefficient at computing each timestep in large networks; explicit methods can compute a timestep efficiently, but timesteps this large are unthinkable with a normal explicit algorithm because they are typically unstable. In this paper we shall demonstrate explicit integration methods that realize such large integration timesteps in a variety of extremely stiff examples. This alters essentially the discussion of whether explicit methods, with their faster computation of timesteps and more favorable scaling with network size, are practical for large, stiff networks. The general approximations to be discussed have been implemented before [1,10–13], but we shall find that our implementations are much more successful than previous applications to extremely stiff networks; accordingly we shall reach rather different conclusions about these methods than those of earlier publications.

4. Algebraic stabilization of solutions far from equilibrium

Let us address first explicit approximations that are appropriate if the system is far from microscopic equilibrium. (Methods to determine whether this condition is satisfied will be discussed in Section 5.)

4.1. Explicit asymptotic approximations

The differential equations to be solved take the form given by Eq. (1). Generally, F_i^+ and F_i^- for a given species i each consist of a number of terms depending on the other populations in the network. For the networks that we shall consider the depletion flux for the population y_i will be proportional to y_i,

$$F_i^- = \left(k_1^i + k_2^i + \cdots + k_m^i \right) y_i \equiv k^i y_i, \tag{2}$$

where the k_j^i are rate parameters (in units of time^{-1}) for each of the m processes that can deplete y_i, which may depend on the populations y_j and on system variables such as temperature and density. The characteristic timescales $\tau_j^i = 1/k_j^i$ will vary over many orders of magnitude in the systems of interest, meaning that these equations are very stiff. From Eq. (2) the effective total depletion rate k^i for y_i at a given time, and a corresponding timescale τ^i, may be defined as

$$k^i \equiv \frac{F_i^-}{y_i} \quad \tau^i = \frac{1}{k^i}, \tag{3}$$

permitting Eq. (1) to be written as

$$y_i = \frac{1}{k^i} \left(F_i^+ - \frac{dy_i}{dt} \right). \tag{4}$$

Thus, in a finite-difference approximation at timestep t_n

$$y_i(t_n) = \frac{F_i^+(t_n)}{k^i(t_n)} - \frac{1}{k^i(t_n)} \frac{dy_i}{dt} \bigg|_{t=t_n}. \tag{5}$$

The *asymptotic limit* for the species i corresponds to the condition $F_i^+ \simeq F_i^-$, implying from Eq. (1) that $dy_i/dt \simeq 0$. In this limit Eq. (5) gives a first approximation $y_i^{(1)}(t_n)$ and local error $E_n^{(1)}$, respectively, for $y_i(t_n)$,

$$y_i^{(1)}(t_n) = \frac{F_i^+(t_n)}{k^i(t_n)} \quad E_n^{(1)} \equiv y(t_n) - y^{(1)}(t_n) = -\frac{1}{k(t_n)} \frac{dy}{dt}(t_n). \tag{6}$$

For small dy_i/dt a correction term may be obtained by writing the derivative term in Eq. (5) as

$$\frac{dy}{dt}(t_n) = \frac{1}{\Delta t} (y_i(t_n) - y_i(t_{n-1})) + \frac{1}{\Delta t} \left(E_n^{(1)} - E_{n-1}^{(1)} \right) + O(\Delta t), \tag{7}$$

where $O(x)$ denotes terms of order x. Substitution in Eq. (5) then gives

$$y_n^{(2)} = \frac{F_n^+}{k_n} - \frac{1}{k(t_n)\Delta t} (y(t_n) - y(t_{n-1}) - \frac{1}{k(t_n)\Delta t} \left(-E_n^{(1)} + E_{n-1}^{(1)} \right),$$

and setting $y(t_n) = y_n^{(2)}$ and solving for $y_n^{(2)}$ gives

$$y_n^{(2)} = \frac{1}{1 + k_n \Delta t} (y_{n-1} + F_n^+ \Delta t), \tag{8}$$

where a term of order $(\Delta t)^2/(1 + k(t_n)\Delta t)$ has been discarded. This approximation is expected to be valid for large $k\Delta t$. Another approach is to use a predictor–corrector scheme within such an asymptotic approximation [1]. However, it has been shown [9] that these asymptotic approximations give rather similar results for the networks that we shall test, so the simple formula (8) will be adequate for the present discussion. The mathematical and numerical properties of asymptotic approximations have been explored previously in Refs. [1,10–12].

To implement an asymptotic algorithm we define a critical value κ of $k\Delta t$ and at each timestep cycle through all populations and compute the product $k^i\Delta t$ for each species i using Eq. (3) and the proposed timestep Δt. Then, for each species i

(1) If $k^i\Delta t < \kappa$, the population is updated numerically by the explicit Euler method.
(2) If $k\Delta t \geqslant \kappa$, the population is updated algebraically using Eq. (8).

Formally explicit integration is expected to be stable if $k^i\Delta t < 1$ and potentially unstable if $k^i\Delta t \geqslant 1$ (see the discussions in Refs. [1,9]). Therefore, $\kappa = 1$ has been chosen for all examples presented here.

4.2. Quasi-steady-state approximations

An alternative explicit algebraic solution to the coupled differential equations is possible using the quasi-steady-state (QSS) approximations developed by Mott and collaborators [11,12], which drew on earlier work in Refs. [14–16]. Following Mott et al. [11,12], first notice that Eq. (1), expressed in the form $dy/dt = F^+(t) - k(t)y(t)$ using Eq. (2) and with indices dropped for notational convenience, has the general solution

$$y(t) = y_0 e^{-kt} + \frac{F^+}{k}(1 - e^{-kt}), \tag{9}$$

provided that k and F^+ are constant. The QSS method then uses this equation as the basis of a predictor–corrector algorithm in which a prediction is made using only initial values and then a corrector is applied that uses a combination of initial values and values computed using the predictor solution. In terms of a parameter $\alpha(r)$ defined by

$$\alpha(r) = \frac{160r^3 + 60r^2 + 11r + 1}{360r^3 + 60r^2 + 12r + 1}, \tag{10}$$

where $r \equiv 1/k\Delta t$, we adopt a predictor y^p and corrector y^c given by Mott et al. [11,12]

$$y^p = y^0 + \frac{\Delta t(F_0^+ - F_0^-)}{1 + \alpha^0 k^0 \Delta t} \quad y^c = y^0 + \frac{\widetilde{F}^+ - \bar{k}y^0}{1 + \bar{\alpha}\bar{k}\Delta t}, \tag{11}$$

where α^0 is evaluated from Eq. (10) with $r = 1/k^0\Delta t$, an average rate parameter is defined by $\bar{k} = \frac{1}{2}(k^0 + k^p)$, $\bar{\alpha}$ is specified by Eq. (10) with $r = 1/\bar{k}\Delta t$, and

$$\widetilde{F}^+ = \bar{\alpha}F_p^+ + (1 - \bar{\alpha})F_0^+.$$

The corrector can be iterated if desired by using y^c from one iteration step as the y^p for the next iteration step. We implement an explicit QSS algorithm based on the predictor–corrector (11) essentially in parallel with that described above for the asymptotic method, except that in applying the QSS algorithm all equations are treated in QSS approximation, rather than dividing the equations into a set treated by explicit forward difference and a set treated analytically [17].

5. Algebraic stabilization in the approach to microscopic equilibrium

We shall find that the asymptotic and quasi-steady-state methods described in preceding sections work well for macroscopic equilibration, but are highly inefficient for microscopic equilibration. Thus, these methods must be augmented by a means to remove stiffness associated with the approach to (microscopic) equilibrium if they are to be applicable to a broad range of problems. In this section approximations to stabilize explicit integration in the presence of microscopic equilibration are developed. This development draws heavily on the partial equilibrium work of Mott [12,13] as a starting point, but we shall extend these methods and find much more favorable results for extremely stiff networks than those obtained in the pioneering work of Mott and collaborators [18].

Partial equilibrium (PE) methods examine source terms f_i^+ and f_i^- for individual reaction pairs in the network—not the composite fluxes F_i^+ and F_i^- that are the basis for asymptotic and QSS approximations—for approach to equilibrium. When a fast reaction pair nears equilibrium its source terms are removed from the direct numerical integration in favor of an equilibrium algebraic constraint. Reactions not in equilibrium still contribute to the fluxes for the numerical integrator, but once fast reactions are decoupled from the numerical integration the remaining system typically becomes (much) less stiff. Consider a representative 2-body reaction and its source term $f_{ab \rightleftharpoons cd}$,

$$a + b \rightleftharpoons c + d \quad f_{ab \rightleftharpoons cd} = \pm(k_f y_a y_b - k_r y_c y_d), \tag{12}$$

where the y_i denote population variables for the species i and the ks are rate parameters for forward (k_f) and reverse (k_r) reactions. Considered in isolation, the reaction pair of Eq. (12) may be deemed to be in equilibrium if $f_{ab \rightleftharpoons cd} = 0$. It is useful to introduce the idea of *partial equilibrium (PE)*, where at a given time some reaction pairs have $f = 0$ and some have $f \neq 0$. The evolution of the system is then determined primarily by those reactions for which $f \neq 0$, but since the system is coupled the $f \neq 0$ reactions will perturb the $f = 0$ reaction pairs so that for those reactions near equilibrium $f = 0 \rightarrow f \sim 0$. Let us now introduce a set of definitions and concepts that will allow us both to quantify and use to our advantage the partial equilibrium condition $f \sim 0$.

5.1. Conserved scalars and progress variables

The reaction pair $a + b \rightleftharpoons c + d$ appears at first glance to have four characteristic timescales associated with the rate of change for the four populations y_a, y_b, y_c, and y_d, respectively. However, it is clear that the following three constraints apply to this reaction

$$y_a - y_b = c_1 \quad y_a + y_c = c_2 \quad y_a + y_d = c_3, \tag{13}$$

where the constants c_i may be evaluated by substituting the initial abundances into Eq. (13),

$$c_1 = y_a^0 - y_b^0 \quad c_2 = y_a^0 + y_c^0 \quad c_3 = y_a^0 + y_d^0.$$

Losing one a in the reaction $a + b \rightleftharpoons c + d$ requires the simultaneous loss of one b, so their difference must be constant, as implied by the first of Eq. (13), and every loss of one a produces one c and one d, implying the second and third of Eq. (13). The left sides in the equations (13) are examples of *conserved scalars* [12,13], which are constant by virtue of the structure of the equations, not by any particular dynamical assumptions. The differential equation describing the evolution of y_a is

$$\frac{dy_a}{dt} = -k_f y_a y_b + k_r y_c y_d = a y_a^2 + b y_a + c, \tag{14}$$

where Eq. (13) has been used and

$$a \equiv k_r - k_f \quad b \equiv -k_r(c_2 + c_3) + k_f c_1 \quad c \equiv k_r c_2 c_3.$$

We shall demonstrate below that the approach to equilibrium for any 2-body reaction pair can be described, and the approach to equilibrium for any 3-body reaction pair can be approximated, by a differential equation of this form. In terms of the quantity

$$q \equiv 4ac - b^2, \tag{15}$$

the solutions to Eq. (14) of interest in the present context correspond to $a \neq 0$ and $q < 0$, and take the form [12,19]

$$y_a(t) = -\frac{1}{2a}\left(b + \sqrt{-q}\,\frac{1 + \phi \exp(-\sqrt{-q}\,t)}{1 - \phi \exp(-\sqrt{-q}\,t)}\right) \quad \phi \equiv \frac{2a y_0 + b + \sqrt{-q}}{2a y_0 + b - \sqrt{-q}}. \tag{16}$$

The equilibrium solution then corresponds to the limit $t \to \infty$ of Eq. (16),

$$\bar{y}_a \equiv y_a^{eq} = -\frac{1}{2a}(b + \sqrt{-q}). \tag{17}$$

Once y_a has been determined the constraints (13) may be used to determine the other abundances. For example, at equilibrium

$$\bar{y}_b(t) = \bar{y}_a(t) - c_1 \quad \bar{y}_c(t) = c_2 - \bar{y}_a(t) \quad \bar{y}_d(t) = c_3 - \bar{y}_a(t).$$

It is often convenient to introduce a variable λ that is the difference between the values of the y_i at the beginning of the timestep and their current values

$$y_a = y_a^0 - \lambda \quad y_b = y_b^0 - \lambda \quad y_c = y_c^0 + \lambda \quad y_d = y_d^0 + \lambda. \tag{18}$$

The new variable λ is termed a *progress variable* for the reaction characterized by $f_{ab \rightleftharpoons cd}$ and satisfies

$$\frac{d\lambda}{dt} = f_{ab \rightleftharpoons cd} \quad \lambda_0 \equiv \lambda(t = 0) = 0. \tag{19}$$

Thus the approach of $a + b \rightleftharpoons c + d$ to equilibrium is controlled by a single differential equation (14) that can be expressed in terms of either a single one of the abundances y_i, or the progress variable λ. The general solution of this equation is of the form given by Eq. (16), with the time dependence residing dominantly in the exponentials. Therefore, the rate at which $a + b \rightleftharpoons c + d$ evolves toward the equilibrium solution (17) is governed by a *single timescale*

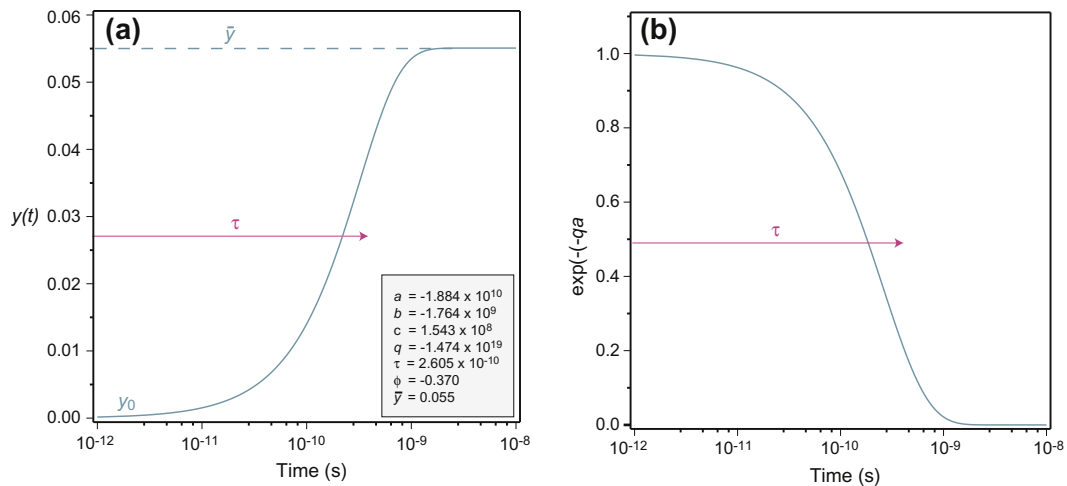$$\tau = \frac{1}{\sqrt{-q}}, \tag{20}$$

which is illustrated in Fig. 2.

Fig. 2. (a) Time evolution of the solution (16) assuming constant a, b, and c. The characteristic timescale for approach to equilibrium (20) is labeled τ and the equilibrium value of $y(t)$ defined by Eq. (17) is denoted by $\bar{y}$. To illustrate we have assumed the initial value $y_0 = 0$. For times considerably larger than τ the general solution (16) saturates at the equilibrium solution (17). (b) Behavior of the exponential factor in Eq. (16).

Whether a reaction pair is near equilibrium at time t may then be determined by requiring that

$$\frac{|y_i(t) - \bar{y}_i|}{\bar{y}_i} < \epsilon_i \tag{21}$$

for each of the species i that is involved in the reaction pair, where $y_i(t)$ is the actual abundance, $\bar{y}_i$ is the equilibrium abundance determined from Eq. (17), and ϵ_i is a user-specified tolerance that could depend on i but will be taken to be the same for all species in the examples discussed to be here. Alternatively, the equilibrium timescale (20) may be compared with the current numerical timestep to determine whether a reaction is near equilibrium: if τ is much smaller than the timestep, we may expect that equilibrium can be established and maintained in successive timesteps, even if it is being continually disturbed by other non-equilibrated processes.

5.2. Reaction vectors

A partial equilibrium approximation in a large network could require that thousands of reactions be examined for their equilibrium status at each timestep. Let us introduce a formalism, adapted from the work of Mott et al. [12,13], that allows examination of partial equilibrium criteria in a particularly efficient way by exploiting the analogy of a reaction network to a linear vector space. This will have two large advantages: (1) it will provide us with some well-established mathematical tools. (2) The abstraction of reaction network as linear vector space permits formulation of a partial equilibrium algorithm that is not strongly tied to the details of a particular problem, thus aiding portability within and across disciplines.

We begin by expressing the concentration variables for the n species A_i in a network as components of a composition vector

$$\boldsymbol{y} = (y_1, y_2, y_3, \ldots y_n), \tag{22}$$

which lies in an n-dimensional vector space Φ. The components y_i are proportional to number densities for the species labeled by A_i, so a specific vector in this space defines a particular composition. Any reaction in the network can then be written in the form $\sum_{i=1}^{n} a_i A_i \rightleftharpoons \sum_{i=1}^{n} b_i A_i$ for some sets of coefficients $\{a_i\}$ and $\{b_i\}$. The coefficients on the two sides of a reaction may be used to define a vector $\boldsymbol{r} \in \Phi$ with components

$$\boldsymbol{r} \equiv (b_1 - a_1, b_2 - a_2, \ldots, b_n - a_n), \tag{23}$$

that specifies how the composition may change because of the reaction. For example, consider the main part of the CNO cycle illustrated on the right side of Fig. 1. Choosing a basis $\{p, \alpha, {}^{12}\text{C}, {}^{13}\text{N}, {}^{13}\text{C}, {}^{14}\text{N}, {}^{15}\text{O}, {}^{15}\text{N}\}$, the reaction $p + {}^{15}\text{N} \rightarrow \alpha + {}^{12}\text{C}$ then has a reaction vector $\boldsymbol{r}$ with components $(-1, 1, 1, 0, 0, 0, 0, -1)$.

For a network with three or fewer species the corresponding linear vector space can be displayed geometrically. For example, Fig. 3 illustrates a network containing the isotopes $\{{}^4\text{He}, {}^8\text{Be}, {}^{12}\text{C}\}$ and four reaction vectors corresponding to the reaction pairs $2\alpha \rightleftharpoons {}^8\text{Be}$ and $\alpha + {}^8\text{Be} \rightleftharpoons {}^{12}\text{C}$. Larger networks cannot be visualized so easily, but their algebraic properties remain completely analogous to those of a simple network like that illustrated in Fig. 3.
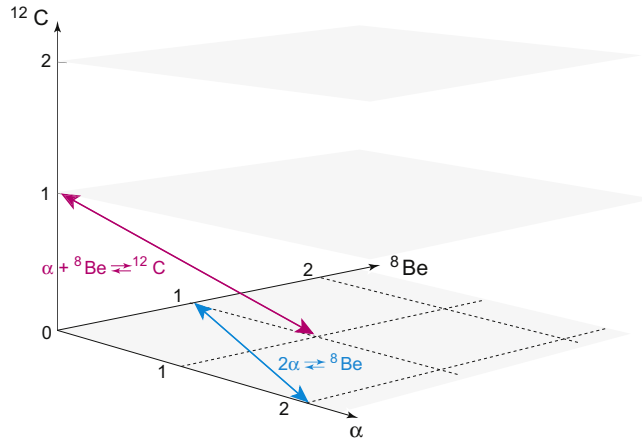
Fig. 3. Four reaction vectors for a simple 3-species network corresponding to the astrophysical triple-α process that converts ^{4}He to ^{12}C in red giant stars. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

5.3. Conservation laws

Given an initial composition $\mathbf{y}_0 = (y_1^0, y_2^0, \ldots y_n^0)$, a single pair of reactions labeled by i can produce a composition $\mathbf{y} = \mathbf{y}_0 + \alpha_i \mathbf{r}_i$, so for a set of k possible reactions

$$\mathbf{y} = \mathbf{y}_0 + \sum_{i=1}^{k} \alpha_i \mathbf{r}_i. \tag{24}$$

Define a time-independent vector $\mathbf{c} \in \Phi$ that is orthogonal to each of the k vectors $\mathbf{r}_i$ in Eq. (24). Then from Eq. (24), $\mathbf{c} \cdot \mathbf{y} = \mathbf{c} \cdot \mathbf{y}_0$ and therefore

$$\sum_{i=1}^{n} c_i y_i = \sum_{i=1}^{n} c_i y_0^i = \text{constant}, \tag{25}$$

so any vector $\mathbf{c}$ orthogonal to the reaction vectors $\mathbf{r}_1, \mathbf{r}_2, \ldots, \mathbf{r}_k$ gives a linear combination of abundances that is invariant under all actions of the network. Eq. (25) is a *conservation law of the system* that follows purely from the structure of the network and thus is valid irrespective of dynamical conditions in the network. The conserved quantities may be determined by solving the equation $\mathbf{r} \cdot \mathbf{c} = 0$, where $\mathbf{r}$ is a matrix with rows composed of all reaction vectors (23), for the vectors $\mathbf{c}_k$. For the thermonuclear reaction networks to be used as illustration here, only the nucleon number (sum of protons and neutrons) is typically conserved, since the nuclear and weak reactions can alter isotopic composition but generally cannot change the baryon number. (We assume that energies are sufficiently low that neutrons and protons are the only baryons in the system.) In more general reaction networks involving much lower energy scales such as in chemical kinetics, there may be a number of additional conservation-law constraints, such as those protecting the integrity of atoms in the reactions. Such cases have been discussed extensively by Mott and collaborators [12,13].

Thus, once a basis has been chosen for a specific reaction network it is straightforward to enumerate all of its possible compositions in terms of a set of vectors $\mathbf{y}_i$, all possible reactions in terms of a set of vectors $\mathbf{r}_j$, and the conservation laws that follow from the structure of the network in terms of a set of vectors $\mathbf{c}_k$. This abstract description of a reaction network retains a reference to a specific physical problem only through the choice of basis and the choice of allowed reaction vectors.

5.4. Reaction group classes

It will prove useful to associate inverse reaction pairs in *reaction group classes* (*reaction groups,* or *RG* for short), in which all reactions of a group share the same reaction vector $\mathbf{r}$, up to a sign. To illustrate we employ the reaction classifications used in the REACLIB library [20] that are illustrated in Table 1. In this classification the reactions of importance in nuclear astrophysics are assigned to eight categories, depending on the number of *nuclear species* on the two sides of the reaction equation. From this classification we see that there are five independent ways to combine the reactions of Table 1 into reversible reaction pairs, leading to the reaction group classification illustrated in Table 2. For example, reaction group B consists of reactions from REACLIB reaction class 2 ($a \to b + c$) paired with their inverse reactions ($b + c \to a$), which belong to REACLIB reaction class 4.

For each reaction group class the characteristic differential equation governing the reaction pair considered in isolation is of the form given by Eq. (14), $dy/dt = ay^2 + by + c$, where y is either an abundance variable (proportional to a number density) for one of the reaction species, or a progress variable measuring the change from initial abundances associated with the

Table 1
Reaction classes in the REACLIB [20] library.

Class	Reaction	Description or example
1	$a \to b$	β-decay or e^- capture
2	$a \to b + c$	Photodisintegration + α
3	$a \to b + c + d$	$^{12}C \to 3\alpha$
4	$a + b \to c$	Capture reactions
5	$a + b \to c + d$	Exchange reactions
6	$a + b \to c + d + e$	$^2H + {}^7Be \to {}^1H + 2{}^4He$
7	$a + b \to c + d + e + f$	$^3He + {}^7Be \to 2{}^1H + 2{}^4He$
8	$a + b + c \to d \ (+e)$	Effective 3-body reactions

Table 2
Reaction group classes.

Class	Reaction pair	REACLIB class pairing
A	$a \rightleftharpoons b$	1 with 1
B	$a + b \rightleftharpoons c$	2 with 4
C	$a + b + c \rightleftharpoons d$	3 with part of 8
D	$a + b \rightleftharpoons c + d$	5 with 5
E	$a + b \rightleftharpoons c + d + e$	6 with part of 8

reaction pair, and the coefficients a, b, and c are known parameters depending on the reaction rates that will be assumed constant within a single network timestep. An exception occurs for reaction group classes C and E, where there are a 3-body reactions and the most general form of the differential equation involves cubic terms, $dy/dt = \alpha y^3 + \beta y^2 + \gamma y + \epsilon$. These "3-body" reactions in astrophysics are typically actually sequential 2-body reactions and we assume that in any timestep $y(t)^3 \simeq y^{(0)}y(t)^2$, where $y^{(0)}$ is the (constant) value of $y(t)$ at the beginning of the timestep. This reduces the cubic equation to an effective quadratic equation of the form (14), with $a = \alpha y^{(0)} + \beta$, $b = \gamma$, and $c = \epsilon$. This has been found to be a very good approximation in typical astrophysical environments and we apply it to all 3-body reactions in the examples discussed here.

5.5. Equilibrium constraints

If a reaction pair from a specific reaction group class is near equilibrium, there will be a corresponding equilibrium constraint that takes the general form [11]

$$\prod_{i=1}^{n} y_i^{(b_i - a_i)} = K, \tag{26}$$

where K is a ratio of rate parameters. For example, consider the reaction group class E pair $a + b \rightleftharpoons c + d + e$ in isolation, with differential equations for the populations y_i written in the form

$$\dot{y}_a = \dot{y}_b = -\dot{y}_c = -\dot{y}_d = -\dot{y}_e = -k_f y_a y_b + k_r y_c y_d y_e.$$

Then at equilibrium, requiring that the forward flux $-k_f y_a y_b$ and reverse flux $k_r y_c y_d y_e$ in the reaction pair sum to zero gives the constraint

$$\frac{y_a y_b}{y_c y_d y_e} = \frac{k_r}{k_f} \equiv K,$$

which is of the form (26).

5.6. Systematic classification of reaction group properties

Applying the principles discussed in the preceding paragraphs to the reaction group classes in Table 2 gives the results summarized for reaction group classes A–E in Appendix A. This reaction group classification has been developed assuming astrophysical thermonuclear networks and a particular parameterization (REACLIB) of the corresponding reaction rates. However, the illustrated methodology is of wider significance. First, since any reaction compilation in astrophysics could be reparameterized in the REACLIB format, this classification scheme provides a general partial-equilibrium bookkeeping that could be applied to any problem in astrophysics. Second, for any large reaction network in any field we may apply the classification techniques illustrated here to group all reactions of the network into reaction group classes and deduce for each reaction group class analytical expressions for all quantities necessary for applying a PE approximation. All that is required is to cast the network as a linear algebra problem by choosing a set of basis vectors corresponding to the species of the network, and then to define the corresponding reaction vectors of physical importance within this space. Once that is

done, the formalism developed here may be applied systematically. In principle this classification need be developed only once for the networks of importance in any particular discipline, as we have illustrated here specifically for astrophysics.

5.7. General methods for partial equilibrium calculations

We now have a set of tools to implement partial equilibrium approximations, but there are some practical issues to resolve before it is possible to make realistic calculations. To address those issues, let us now outline a specific application approach. Although our remarks will be illustrated by examples using astrophysical thermonuclear networks, the methods discussed should be relevant for a much broader range of problems.

5.7.1. Overview of approach

We employ the partial equilibrium method in conjunction with the asymptotic approximation. (There may be advantages in using quasi-steady-state methods instead of asymptotic ones, but we shall deal with that in future work.) Once the reactions of the network are classified into reaction groups, the algorithm has three steps:

(1) A numerical integration step begins with the full network of differential equations, but in computing fluxes all terms involving reaction groups judged to be equilibrated (based on criteria determined by species populations at the end of the previous timestep) are assumed to sum identically to zero net flux and are omitted from the flux summations.
(2) A timestep Δt is chosen and used in conjunction with the fluxes to determine which species satisfy the asymptotic condition. For those species that are not asymptotic, the change in abundance for the timestep is then computed by ordinary forward (explicit) finite difference, but for those species that are judged to be asymptotic the new abundance for the timestep is instead computed analytically using the asymptotic approximation.
(3) For all species in reaction groups judged to be equilibrated at the beginning of the timestep, it is assumed that reactions not in equilibrium will have driven these populations slightly away from equilibrium during the timestep. These populations are then adjusted, subject to the system's conservation laws, to restore their equilibrium values at the end of the timestep.

Hence the partial equilibrium part of the approximation does not reduce the number of differential equations integrated numerically within a timestep, but instead removes systematically the stiffest parts of their fluxes. In contrast, the asymptotic approximation effectively reduces the number of differential equations integrated numerically by replacing the numerical forward difference with an analytic asymptotic abundance for those isotopes satisfying the asymptotic condition.

The partial equilibrium and asymptotic approaches are complementary because partial equilibrium can operate microscopically to make the differential equation for a given isotope less stiff, even if that isotope does not satisfy the asymptotic condition, while the asymptotic condition removes entire differential equations from the numerical update and thus can operate macroscopically to remove stiff reaction components, even if they do not satisfy partial equilibrium conditions. For brevity we shall often refer simply to the partial equilibrium (PE) approximation, but this always means the partial equilibrium approximation used in conjunction with the asymptotic approximation, in the manner just described.

5.7.2. Complications in realistic networks

The complication for the basic approach outlined in Section 5.7.1 when applied to a realistic network is that there may be more than one computed equilibrium value for a given species. This is because the equilibrium abundance from Eq. (17) is to be computed separately for each reaction group, and a given isotope often will be found in more than one such group. In the general case these computed equilibrium values can be different, since equilibrium within each reaction group is specified only to the tolerance implied by ϵ_i in Eq. (21). However, the equilibrium abundances of a given species computed for each equilibrated reaction group that it is a member of cannot differ substantially among themselves, since this would contradict the evidence supplied by the network that the reaction groups are in equilibrium. For a species i there is only *one* actual abundance y_i in the network at a given time, which must satisfy *simultaneously* the equilibrium conditions for all reaction groups determined to be in equilibrium, within the tolerances of Eq. (21).

Therefore, restoration of equilibrium at the end of a numerical integration timestep will correspond to setting the abundance of each isotope to a compromise choice among each of the (similar) predicted equilibrium values for all equilibrated reaction groups in which it participates. This is a self-consistent approximation if the variation in possible equilibrium abundances remains less than the tolerances ϵ_i used to impose equilibrium in Eq. (21). For the networks that we have tested, this seems generally to be fulfilled for appropriate choices of ϵ_i.

5.7.3. Specific methods for restoring equilibrium

We have investigated three methods for restoring equilibrium at the end of the numerical timestep:

(1) Reimpose equilibrium ratios (Eq. (26)) by Newton–Raphson iteration.
(2) Reimpose equilibrium abundances (Eq. (17)) by Newton–Raphson iteration.
(3) Reimpose equilibrium abundances (Eq. (17)) by averaging over progress variables.

All three methods are described in Ref. [18], where we show that for the examples investigated to date the third method gives essentially the same results as the first two and is considerably simpler to implement, since it involves no matrices and no iteration. Thus we outline only this method and refer the reader to Ref. [18] for the technical details on it and the other two methods.

The basic idea is that in partial equilibrium the isotopic abundances in a reaction group evolve according to a single time-scale given by Eq. (20), as discussed in Section 5.1. Thus, within a single reaction group, the equilibrium abundance of any one isotope, or the progress variable for the reaction group, determines the equilibrium abundance of all species in the group. Furthermore, within a single reaction group the evolution of all species to equilibrium conserves particle number, by virtue of constraints such as those of Eq. (13) that are summarized for all five reaction group classes in Appendix A. Let us exploit this using the progress variable λ_i from each reaction group. If all the reaction groups were independent, then equilibrium could be restored by requiring for all reaction groups in equilibrium $\lambda_i - \bar{\lambda}_i = 0$, where $\bar{\lambda}_i$ denotes the equilibrium value of λ_i computed from Eq. (17) and relations like Eq. (18). Once the equilibrium value of λ_i is computed, the equilibrium values for all other isotopes in the group then follow from constraints like Eq. (18) that are tabulated for all reaction group classes in Appendix A.

The simple considerations of the preceding paragraph are insufficient because the reaction groups are generally *not* independent: the individual network species may appear in more than one reaction group, as discussed in Section 5.7.2. The equilibrium condition implies that we must have approximately equal computed equilibrium values for an isotope participating in more than one RG, but exact equality does not hold because of the finite tolerance ϵ_i used to test for equilibrium in Eq. (21). Thus, we restore equilibrium for each isotope participating in partial equilibrium at the end of a timestep by replacing its numerically-computed abundance with its equilibrium value (17), averaged arithmetically over all equilibrated reaction groups in which it participates.

There is one final issue: evolution to equilibrium for individual reaction groups considered in isolation conserves particle number, but the averaging procedure introduces a small fluctuation since the average will generally differ from the individual $\bar{y}_i$ that were computed conserving particle number. Thus, for each timestep, after equilibrium has been restored, we rescale all y_i by a multiplicative factor that restores the total nucleon number to its value at the start of the timestep.

6. A simple adaptive timestepper for explicit integration

For testing the asymptotic and QSS methods a simple adaptive timestepper has been used that is described in more detail in Ref. [9]:

(1) Compute a trial integration timestep based on limiting the change in populations that would result from that timestep to a specified tolerance. Choose the minimum of this trial timestep and the timestep taken in the previous integration step and update all populations by the asymptotic or QSS algorithms.
(2) Check for conservation of particle number within a specified tolerance range. If not satisfied, increase or decrease the timestep as appropriate by a small factor and repeat the timestep using the original fluxes.

Our asymptotic plus PE calculations use a modified form of this timestepper that de-emphasizes the second step, since the PE algorithm itself implements approximate probability conservation. This prescription is far from optimized, but it gives stable and accurate results for the varied astrophysical networks that have been tested. Thus it is adequate for our primary task here, which is to establish whether explicit methods can even compete with implicit methods for stiff networks.

7. Comparisons of explicit and implicit integration speeds

In the following examples explicit and implicit integration speeds will be compared using algorithms at different stages of development and optimization. Thus, the codes cannot yet be simply compared head to head. We shall make a simple assumption that for codes at similar levels of optimization the primary difference between explicit and implicit methods would be in the extra time spent in matrix operations for the implicit method. Thus, if the fraction of computing time spent in the linear algebra is f for an implicit code, it will be assumed that an explicit code at a similar level of optimization could compute a timestep faster by a factor of $F = 1/(1 - f)$. We shall then assume that for implicit and explicit codes at similar levels of optimization the ratio of speeds for a given problem would be F times the ratio of integration steps that are required by the two codes. As discussed later, this may underestimate the speed of a fully-optimized explicit code relative to implicit codes, but establishes a lower limit on how fast the explicit calculation can be.

Estimated factors F are shown in Table 3 for the networks to be discussed, based on data obtained by Feger [19] using the implicit, backward-Euler code Xnet [21] with both dense solvers (LAPACK [22]) and sparse solvers (MA28 [23] and PARDISO [24]), assuming the optimal solver to be used by the implicit code for a given network. We see that for very small networks implicit and explicit methods require similar times to compute a timestep, but for larger networks the explicit computation of a timestep can be faster than that for the implicit code by a factor of 10 or more for the networks examined here.

Table 3
Speedup factors for explicit vs. implicit timesteps [19].

Network	Isotopes	Speedup F
pp	7	$\sim$1.5
Alpha	16	3
Nova	134	7
150-isotope	150	7.5
365-isotope	365	$\sim$20

8. Tests of asymptotic and QSS algorithms

This section presents some representative calculations for astrophysical thermonuclear networks using the explicit asymptotic (Asy) and quasi-steady-state (QSS) methods. For astrophysical networks we shall replace the generic y_i of Eq. (1) with population variables common for astrophysics: the mass fraction X_i or the (molar) abundance Y_i, with

$$X_i = \frac{n_i A_i}{\rho N_A} \quad Y_i \equiv \frac{X_i}{A_i} = \frac{n_i}{\rho N_A} \tag{27}$$

where N_A is Avogadro's number, ρ is the total mass density, A_i is the atomic mass number, n_i the number density for the species i, and conservation of nucleon number requires $\sum X_i = 1$.

8.1. Explicit asymptotic and QSS integration of the pp-chains

The proton–proton chains (pp-chains) that convert hydrogen to helium and thereby power the Sun provide a spectacular illustration of stiffness in a simple yet physically-important network. Fig. 4 exhibits integration of the pp-chains at a constant temperature and density characteristic of the current core of the Sun using the asymptotic method, the QSS method, and the implicit backward-Euler code Xnet [21]. We see that the asymptotic and QSS integrations give results for the mass fractions in rather good agreement with the implicit code over 20 orders of magnitude, and generally take timesteps $dt \sim 0.1\,t$ that are comparable to those for the implicit code over the entire range of integration. (The asymptotic method required 333 total integration steps, the QSS method required 286 steps, and the implicit method required 176 steps for this example.) The maximum stable timestep for a standard explicit integration method, which typically may be approximated by the inverse of the fastest rate in the system, is illustrated by the dashed blue curve in Fig. 4(b). Thus, at the end of the calculation the explicit integration timesteps are $\sim 10^{21}$ times larger than would be stable in a normal explicit integration. The calculation illustrated in Fig. 4 takes a fraction of a second on a 3 GHz processor with the asymptotic, QSS, or implicit methods. In contrast, from Fig. 4(b) we estimate that a standard explicit method would require $\sim 10^{21}$ s of processor time to compute the pp-chains to hydrogen depletion, which is a thousand times longer than the age of the Universe.

This example is a simple one, but the results call into question most of what has been said in the literature concerning the use of explicit methods for stiff systems. This network is about as stiff as one will find in any scientific application, with a maximum integration timestep that is 21 orders of magnitude larger than the inverse of the fastest rate in the system. Yet both the explicit asymptotic and explicit quasi-steady-state methods have integrated it with an efficiency and accuracy comparable to that of a standard implicit algorithm.

8.2. Nova explosions

The network of the preceding example was extremely stiff, but small, and involved very stable thermonuclear burning under conditions of hydrostatic equilibrium. Let us now examine a case involving an extremely stiff but much larger network
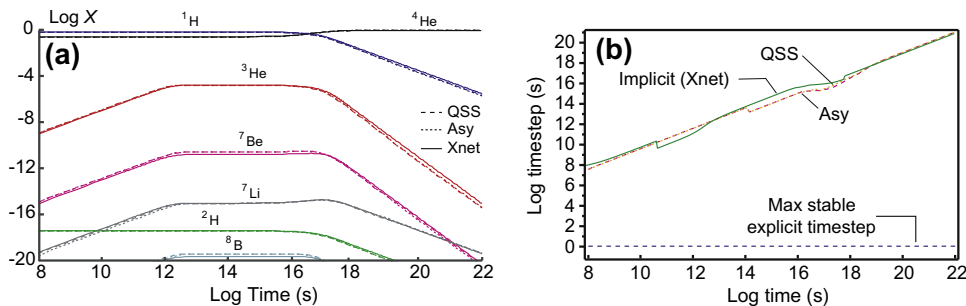

Fig. 4. Integration of the pp-chains at constant temperature $T_9 = 0.016$ (where T_9 denotes temperature in units of 10^9 K) and constant density 160 g cm^{-3}, assuming solar initial abundances. Reaction rates were taken from the REACLIB library [20]. (a) Mass fractions for the asymptotic method, the QSS method, and for the standard implicit code Xnet [21]. (b) Integration timesteps.

in which the burning is highly explosive. In binary star systems for which one star is a white dwarf and the other is a more normal star with a hydrogen envelope, in some circumstances the hydrogen from the companion star may accrete onto the surface of the white dwarf until a thermonuclear explosion is triggered that blows off a thin, rapidly-burning shell of accreted matter. This typically occurs under conditions governed by a degenerate-electron equation of state, which means that the explosion occurs as a thermonuclear runaway in matter that initially does not expand while its temperature rises quickly. This mechanism for explosive thermonuclear burning of hydrogen is termed a *nova*.

Fig. 5(a) illustrates a calculation using the explicit QSS algorithm with a hydrodynamical profile [26] displayed in Fig. 5(b) that is characteristic of a nova outburst. Since there are so many mass-fraction curves we do not attempt to display a direct comparison with an asymptotic or implicit calculation, but note that the agreement is rather good, and the total integrated energy release corresponding to the simulation of Fig. 5 was within 1% of that found for the same network using the explicit asymptotic approximation.

The integration timesteps for the calculation in Fig. 5(a) are displayed in Fig. 6(a). Once burning commences, the QSS and asymptotic solvers take timesteps that are from 10^6 to 10^{10} times larger than would be stable for a normal explicit integration. The explicit QSS timesteps illustrated in Fig. 6(a) are somewhat larger than those of the asymptotic solver, and comparable to or greater than those for a typical implicit code: in this calculation the implicit method required 1332 integration steps, the asymptotic calculation required 935 steps, and the QSS method required 777 steps.

Given that for a network with 134 isotopes the explicit codes should be able to calculate an integration timestep about 7 times faster than the implicit code because they avoid the manipulation of large matrices (Table 3), these results suggest that the explicit QSS algorithm is capable of calculating the nova network about 12 times faster and the explicit asymptotic algorithm about 10 times faster than a state-of-the-art implicit code. This impressive integration speed for both the QSS and asymptotic methods applied to a large, extremely stiff network is possible because few reactions reach equilibrium during the simulation, as determined from Eq. (21) and illustrated in Fig. 6(b).

The ability of explicit methods to deal effectively with large networks under nova conditions is supported further in work reported by Feger et al. [19,27]. There results similar the present ones were demonstrated with asymptotic methods for a nova simulation using a different network, different hydrodynamical profile, and different reaction library than those employed here.

8.3. Tidal supernova simulation

If intermediate-mass black holes ($\sim$1000 solar masses) exist in globular star clusters, a white dwarf passing sufficiently close to one could be strongly tidally distorted, compressing the center sufficiently to ignite a thermonuclear detonation in
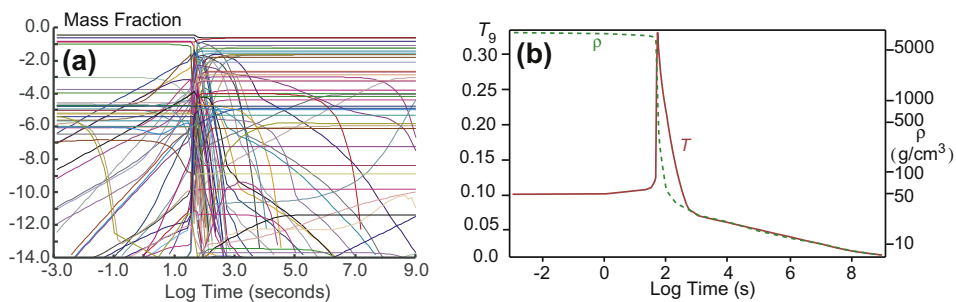

Fig. 5. (a) Mass fractions for a network under nova conditions, corresponding to the hydrodynamical profile shown in (b). The calculation used the QSS method and a network containing 134 isotopes coupled by 1531 reactions, with rates taken from the REACLIB library [20] and initial abundances enriched in heavy elements [25]. The mass density is ρ and T_9 denotes temperature in units of 10^9 K.
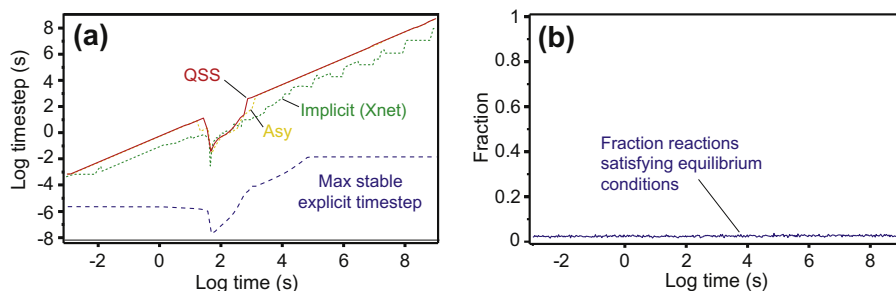

Fig. 6. (a) Timesteps for integration of Fig. 5. (b) Fraction of reactions that reach partial equilibrium in the QSS calculation of Fig. 5.
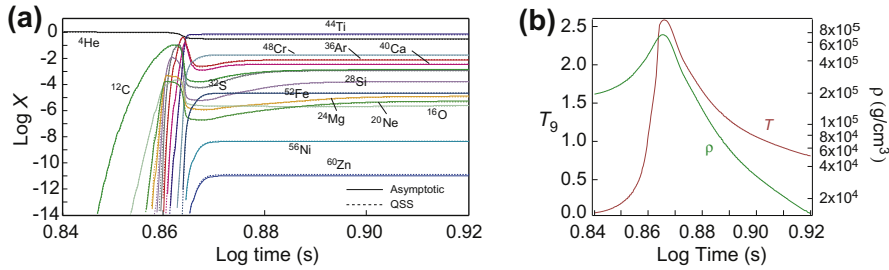
Fig. 7. Comparison of asymptotic and QSS approximation calculations for an alpha network calculation under tidal supernova conditions. Initial composition was pure ^{4}He and REACLIB rates [20] were used. (a) Mass fractions. (b) Hydrodynamical profile [28]. The mass density is ρ and T_9 denotes temperature in units of 10^9 K.
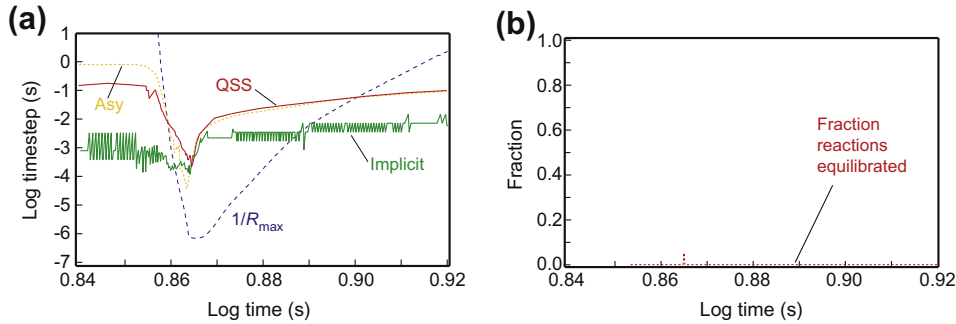

Fig. 8. (a) Integration timesteps and maximum stable purely-explicit timestep $\sim 1/R_{max}$ for the calculation in Fig. 7. (b) Fraction of isotopes that become asymptotic and fraction of reactions equilibrated in the network for the calculation in Fig. 7.
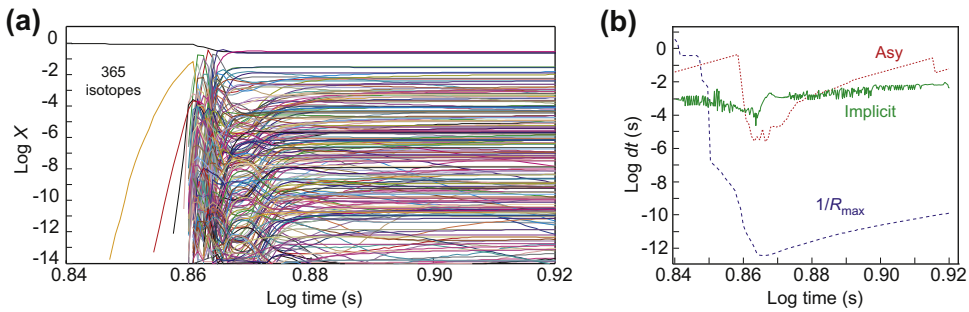

Fig. 9. (a) Mass fractions calculated in asymptotic approximation using asymptotic methods for a 365-element network with 4325 reaction couplings under tidal supernova conditions, corresponding to the hydrodynamical profile shown in Fig. 7(b). Rates were taken from REACLIB [20] and the initial composition was pure ^{4}He. (b) The corresponding integration timesteps and maximum stable explicit timestep $\sim 1/R_{max}$.

the white dwarf under degenerate-electron conditions. Simulations indicate that such *tidal supernovae* would burn carbon and oxygen to iron-group nuclei in a violent thermonuclear runaway that consumes the white dwarf [28].

A comparison of asymptotic and QSS mass fractions and timesteps is shown in Fig. 7(a) for an alpha network with a hydrodynamical profile illustrated in Fig. 7(b) that is characteristic of a supernova induced by tidal interactions in a white dwarf [28]. The integration timestepping is compared for QSS, an asymptotic calculation, and the implicit code Xnet in Fig. 8(a). We see that the timestepping for the QSS calculation is somewhat better than for the asymptotic code and considerably better than for the implicit code (242 total integration steps for the QSS calculation, 480 steps for the asymptotic calculation, and 1185 steps for the implicit calculation). Since from Table 3 an optimized explicit code could compute timesteps about 3 times as fast as an implicit code for an alpha network, we may estimate that the QSS code is capable of calculating this network perhaps 15 times faster, and the asymptotic code perhaps 7 times faster, than the implicit code. The relatively good timestepping for the QSS and asymptotic methods in this case again is primarily because almost no reactions in the network come into equilibrium over the full range of the calculation, as illustrated in Fig. 8(b).

A calculation for the hydrodynamical profile illustrated in Fig. 7(b) for a 365-isotope network using asymptotic and implicit methods is illustrated in Fig. 9. The implicit calculation required 1455 integration steps, compared with 5778 steps for the asymptotic calculation. But Table 3 indicates that for this 365-isotope network the explicit code can calculate each time-step about 20 times faster than the implicit code, so an optimized asymptotic code should be capable of performing the integration in Fig. 9 perhaps 5 times faster than a state-of-the-art implicit code.

This ability of explicit methods to deal effectively with large networks under tidal supernova conditions is supported further by results from Refs. [19,27]. Although different networks and different reaction network rates were used in these references, the explicit asymptotic method was again found to be highly competitive with standard implicit methods for the tidal supernova problem.

8.4. Non-competitive explicit timesteps in the approach to equilibrium

The examples shown to this point have deliberately emphasized examples from networks in which few reactions have become microscopically equilibrated. For such cases the estimated integration speed for optimized quasi-steady-state and asymptotic explicit methods is often comparable to, and in some cases may exceed, that for current implicit codes. Let us now turn to an example representative of a whole class of networks where this is decidedly not true. In a *Type Ia supernova,* a carbon–oxygen white dwarf undergoes runaway thermonuclear burning to iron-group nuclei under degenerate-electron
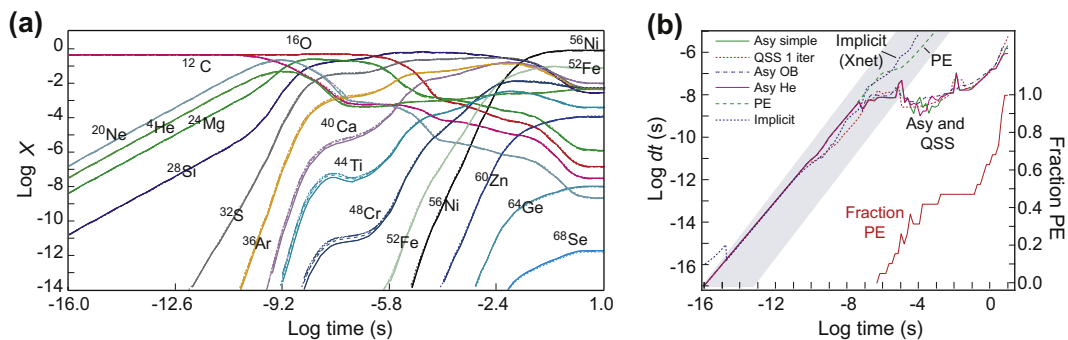

Fig. 10. Comparison of asymptotic and QSS approximations for an alpha network with constant temperature $T_9 = 5$ and constant density of 10^8 g cm^{-3}, using REACLIB rates [20] and initial equal mass fractions of ^{12}C and ^{16}O. (a) Isotopic mass fractions. (b) Integration timesteps (left axis) and fraction of reaction in partial equilibrium (right axis). The gray shaded area represents roughly the region that the explicit timestep profile must lie into have a chance to compete with implicit methods. The different asymptotic methods are labeled Asy and are described in Ref. [9]. The dashed green line (PE) represents the timestepping using the partial equilibrium methods that will be discussed in Section 9. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

Table 4
Reactions of the alpha network for partial equilibrium calculations ($\alpha \equiv {}^4$He). The reverse reactions such as ^{20}Ne $\rightarrow \alpha + {}^{16}$O are photodisintegration reactions, $\gamma + {}^{20}$Ne $\rightarrow \alpha + {}^{16}$O, with the photon γ suppressed in the notation since we track only nuclear species in the network.

Group	Class	Reactions	Members
1	C	$3\alpha \rightleftharpoons {}^{12}$C	4
2	B	$\alpha + {}^{12}$C $\rightleftharpoons {}^{16}$O	4
3	D	^{12}C $+ {}^{12}$C $\rightleftharpoons \alpha + {}^{20}$Ne	2
4	B	$\alpha + {}^{16}$O $\rightleftharpoons {}^{20}$Ne	4
5	D	^{12}C $+ {}^{16}$O $\rightleftharpoons \alpha + {}^{24}$Mg	2
6	D	^{16}O $+ {}^{16}$O $\rightleftharpoons \alpha + {}^{28}$Si	2
7	B	$\alpha + {}^{20}$Ne $\rightleftharpoons {}^{24}$Mg	4
8	D	^{12}C $+ {}^{20}$Ne $\rightleftharpoons \alpha + {}^{28}$Si	2
9	B	$\alpha + {}^{24}$Mg $\rightleftharpoons {}^{28}$Si	4
10	B	$\alpha + {}^{28}$Si $\rightleftharpoons {}^{32}$S	2
11	B	$\alpha + {}^{32}$S $\rightleftharpoons {}^{36}$Ar	2
12	B	$\alpha + {}^{36}$Ar $\rightleftharpoons {}^{40}$Ca	2
13	B	$\alpha + {}^{40}$Ca $\rightleftharpoons {}^{44}$Ti	2
14	B	$\alpha + {}^{44}$Ti $\rightleftharpoons {}^{48}$Cr	2
15	B	$\alpha + {}^{48}$Cr $\rightleftharpoons {}^{52}$Fe	2
16	B	$\alpha + {}^{52}$Fe $\rightleftharpoons {}^{56}$Ni	2
17	B	$\alpha + {}^{56}$Ni $\rightleftharpoons {}^{60}$Zn	2
18	B	$\alpha + {}^{60}$Zn $\rightleftharpoons {}^{64}$Ge	2
19	B	$\alpha + {}^{64}$Ge $\rightleftharpoons {}^{68}$Se	2

conditions, with part of the burning by a subsonic burn front (deflagration) and part by a supersonic burn front (detonation). This typically occurs under much higher temperature and density conditions than for the novae and tidal-supernova examples previously discussed, which tends to produce a large amount of microscopic equilibration before the system begins to expand.

The calculation in Fig. 10 compares various asymptotic approximations and a QSS calculation with an implicit calculation for an alpha-particle network (see Table 4) at a constant temperature and density that might be found in a representative deflagration-burning zone for a Type Ia supernova simulation. Two important conclusions follow from these results.

(1) All of the QSS and asymptotic cases shown are similar, with integrated final energies (not shown) that lie within 1% of each other and total integration times that lie within 25% of each other.
(2) The asymptotic and QSS methods all give timesteps that are potentially competitive with implicit methods (they lie in the shaded gray area) at earlier times, but fall far behind for times greater than about 10^{-5} s.

The reason for conclusion (2) may be seen from the solid red curve on the right of Fig. 10(b), which represents the fraction of reactions that satisfy partial equilibrium conditions. Asymptotic and quasi-steady-state approximations work very well as long as the network is well-removed from microscopic equilibrium, but as soon as significant numbers of reactions become microscopically equilibrated the asymptotic and QSS timestepping begins to lag by large margins. As we now show, *partial equilibrium methods* must be used in conjunction with asymptotic or QSS methods to recover competitive timestepping in the approach to equilibrium. A preview of those results is displayed in Fig. 10(b). The dashed green curve labeled PE corresponds to an asymptotic plus partial equilibrium approximation that exhibits highly-competitive timestepping relative to that of the implicit calculation, even as the network approaches equilibrium.

9. Tests of partial equilibrium for some thermonuclear alpha networks

The partial equilibrium algorithm of Section 5 has been tested in a variety of thermonuclear alpha networks (nuclear networks containing only isotopes that differ from each other by multiples of the ^{4}He nucleus; see Table 4). This section gives some representative examples of those calculations. Because they are among the toughest numerical problems for reaction networks, we shall concentrate on conditions expected in Type Ia supernova explosions (temperatures from 10^7 to 10^{10} K and densities from 10^7 to 10^9 g cm^{-3}). Since the Type Ia explosion is triggered by a thermonuclear runaway in the degenerate matter of a white dwarf, the reaction network must deal with temperatures that can change at rates in excess of 10^{17} K/s. Such conditions lead to rapid equilibration in systems of extremely high stiffness and represent a demanding test of any numerical integration method. Although our longer-term goal is application of the present methods to larger networks with hundreds or even thousands of isotopes, an alpha network provides a fairly realistic, highly-stiff system for initial tests that has the advantage of being small enough to provide transparency in how the algorithm functions. It should also be noted that the most ambitious calculations to date for astrophysical thermonuclear networks coupled to hydrodynamical simulations have employed alpha networks (or even more schematic ones). Reactions and corresponding reaction groups are displayed in Table 4. REACLIB [20] was used for all rates except that inverse rates for reaction groups 3, 5, and 6 are not included in the standard REACLIB tabulation and were taken from Ref. [29]. For all calculations the equilibrium criteria were imposed using Eq. (21), with a constant value $\epsilon_i = 0.01$ for the tolerance parameter.

9.1. Example at constant temperature and density

A calculation for constant $T_9 = 5$ and density of 1×10^7 g cm^{-3} in an alpha network is shown in Fig. 11. The calculated asymptotic plus partial equilibrium (Asy + PE) mass fractions are compared with those of an implicit code in Fig. 11(a). There are small discrepancies in localized regions, especially for some of the weaker populations, but overall agreement is rather good. Mass fractions down to 10^{-14} are displayed for reference purposes. However, for reaction networks coupled to hydrodynamics only mass fractions larger than say $\sim 10^{-2}$–10^{-3} are likely to have significant influence on the hydrodynamics. Thus, the largest discrepancies between PE and implicit mass fractions in Fig. 11(a), and in the other examples that will be discussed, imply uncertainties in the total mass being evolved by the network that would be irrelevant in a coupled hydrodynamical simulation.

Timestepping for various integration methods is illustrated in Fig. 11(b). The PE calculation required 3941 total integration steps while the implicit code required only 600 steps, but this factor of 6.5 timestepping advantage is offset substantially by the expectation that for a 16-isotope network an explicit calculation should be about 3 times faster than the implicit calculation for each timestep (Table 3). Thus we conclude that for fully-optimized codes the implicit calculation would be perhaps twice as fast for this example. Since the semi-implicit timestepping curve was reproduced from another reference the exact number of integration steps is not available, but a comparison of the curves in Fig. 11(b) suggests that an optimized partial equilibrium code is likely to be at least as fast as the semi-implicit YASS code for this example.

The Asy + PE and implicit timesteps are many orders of magnitude better at late times than those of the purely asymptotic (labeled Asy) and purely quasi-steady-state (labeled QSS) calculations. The reason is clear from Fig. 12. For times later than $\sim \log t = -6$, significant numbers of reaction groups come into equilibrium [as determined by the criteria of Eq. (21)], which
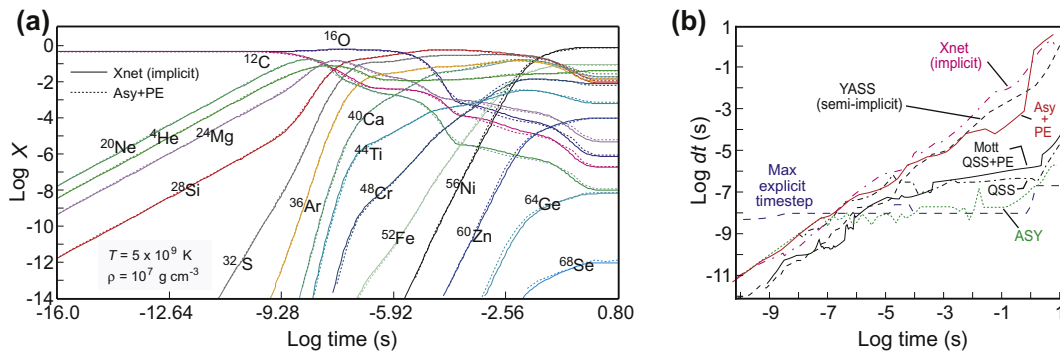
Fig. 11. Calculation for constant $T = 5 \times 10^9$ K ($T_9 = 5$) and $\rho = 1 \times 10^7$ g cm^{-3} alpha network (16 isotopes, 48 reactions, and 19 reaction groups). Initial equal mass fractions of ^{12}C and ^{16}O, and rates from REACLIB [20] and Ref. [29] were used. (a) Mass fractions. (b) Integration timesteps for implicit code Xnet [21], semi-implicit code YASS [30] (reproduced from from Ref. [12]), asymptotic plus PE (present work), QSS plus PE calculation reproduced from Ref. [12], QSS (present work), and asymptotic (present work).
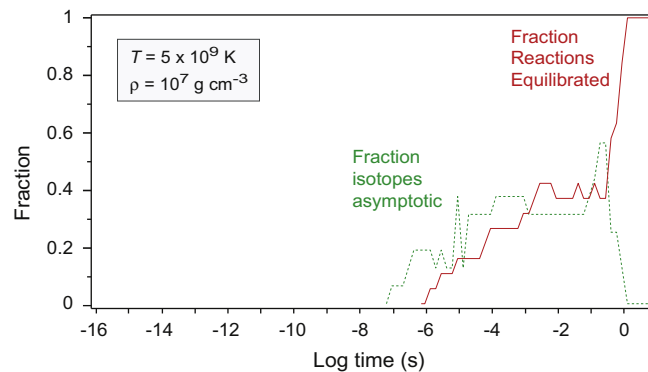

Fig. 12. Fraction of reactions that are treated as being in equilibrium as a function of integration time for the calculation of Fig. 11. Also shown is the fraction of isotopes that become asymptotic in the PE calculation.

asymptotic and QSS methods alone are not designed to handle. The earlier application by Mott [12] of a QSS plus PE calculation for this same system is seen to lag behind the current implementation of asymptotic plus partial equilibrium in time-stepping at late times by a factor of 1000 or more. In fact, the timestepping reproduced from Ref. [12] for QSS + PE is only a little better than that of the pure QSS results from the present paper, and clearly is not competitive with that of either the implicit or semi-implicit calculations, or the present asymptotic plus partial equilibrium result. Thus we see in this example that the huge speed advantage of implicit methods relative to pure asymptotic or quasi-steady-state methods in the approach to equilibrium that was demonstrated in Section 8.4 has been essentially erased by a proper treatment of partial equilibrium in the explicit integrations. In addition, the present implementation of asymptotic plus PE methods is seen to be much faster than previous applications of explicit partial equilibrium methods to this network.

9.2. Example with a hydrodynamical profile

The preceding example employed an alpha network at extreme but constant temperature and density. A partial equilibrium calculation using a hydrodynamical profile with the dramatic temperature rise characteristic of a burning wave in a Type Ia supernova simulation is illustrated in Fig. 13. These are challenging conditions for the reaction network. During the thermonuclear runaway the temperature increases by 3.6 billion K in only 2×10^{-8} s, corresponding to a rate of temperature change 1.7×10^{17} K/s, and the fastest and slowest rates in the network differ by approximately 10 orders of magnitude. Again we see that the partial equilibrium timestepping is competitive with that of the implicit code. The partial equilibrium method required 596 total integration timesteps and the implicit code required 553 steps, so an optimized partial equilibrium calculation would be perhaps three times faster than the corresponding implicit calculation because of its speed advantage in computing each timestep. At late times the timestepping for both the implicit and partial equilibrium calculations is again found to be orders of magnitude larger than that from the purely asymptotic calculation. The reason can be seen from Fig. 13(d): at late times the network is very strongly equilibrated and can be integrated efficiently with an explicit method only if the equilibrating reactions are identified and removed from the numerical integration.
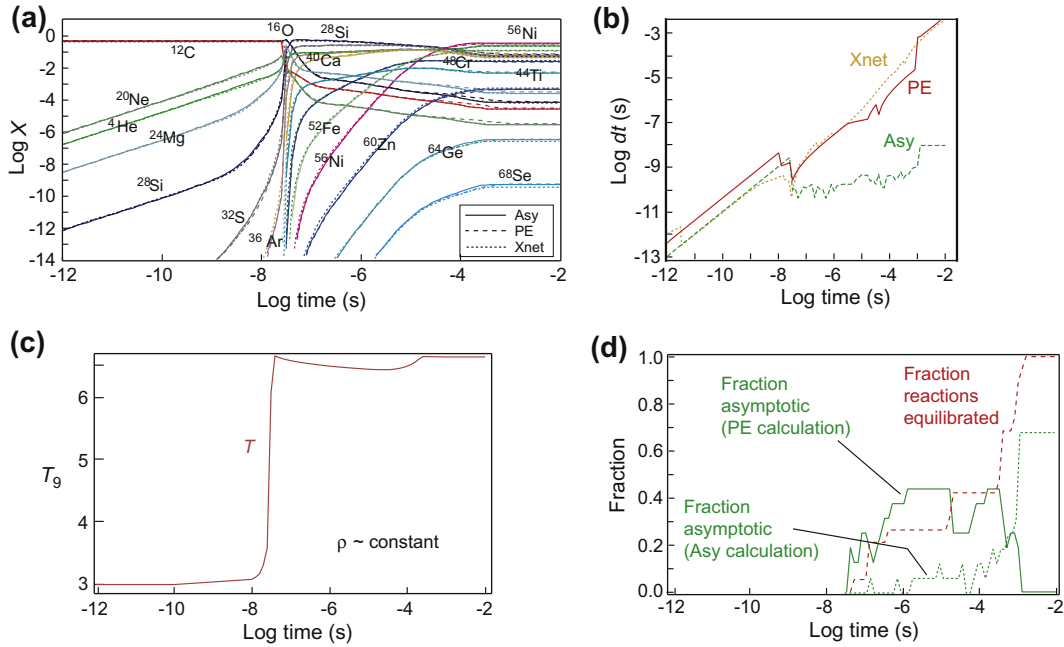
Fig. 13. Asymptotic, partial equilibrium, and implicit calculations for an alpha network under Type Ia supernova conditions. (a) Mass fractions. (b) Integration timesteps. (c) The hydrodynamic profile. (d) Fraction of isotopes that become asymptotic and fraction of network reactions that become microscopicaly equilibrated. Solid green indicates the fraction of asymptotic isotopes in the PE calculation; dashed green indicates the fraction of asymptotic isotopes in the purely asymptotic calculation. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

The synergism of the asymptotic and partial equilibrium approximations is illustrated by Fig. 13(d), where we see that in the asymptotic plus PE calculation isotopes become asymptotic much earlier than in the purely asymptotic calculation. Thus the much larger timestep that the PE plus asymptotic calculation takes relative to the purely asymptotic calculation in the time interval $\log t \sim -7$ to $\log t \sim -3$ is enabled by a combination of reduced stiffness because of the PE approximation and reduced stiffness because more isotopes become asymptotic at early times. This synergism is common and we find typically that replacing the stiffest parts of a network by a complementary set of algebraic constraints exploiting both asymptotic and partial equilibrium conditions is much more effective than either set of constraints used alone.

9.3. Synopsis of partial equilibrium results

The examples shown above are representative of various problems that have been investigated with the new asymptotic plus PE methods. Our general conclusion is that for alpha networks under the extreme conditions corresponding to a Type Ia supernova explosion, the asymptotic plus partial equilibrium algorithm is capable of timestepping that is orders of magnitude better than asymptotic or QSS approximations alone when the system approaches equilibrium. These timesteps typically lie in the same ballpark as those from current implicit and semi-implicit codes, once the partial equilibrium algorithm has removed the fast timescales associated with approach to equilibrium. Because the explicit methods can compute the timestep more efficiently, we find that in many cases they project to be at least as fast as current implicit codes, even for the relatively small networks used as examples here.

10. Scaling with network size

One of the most attractive features of an explicit method is that the time to compute should scale linearly with the size of the network. Because we use modified explicit methods here that require additional bookkeeping to remove stiffness, we should worry about the influence of that additional work on the scaling properties. Since none of the additional bookkeeping requires matrix operations, we surmise on general grounds that this additional overhead should add only incrementally to the computing time and thus that the methods described here should scale linearly with network size.

In Fig. 14 we illustrate scaling of integration time with network size for the nova simulation of Section 8.2 and the tidal supernova simulation of Section 8.3, in asymptotic approximation. The behavior is seen to be approximately linear, as expected. The QSS method requires a comparable amount of overhead as the asymptotic method and so should also be linear. The partial equilibrium method adds some additional computation and logic to the asymptotic or QSS methods but again no
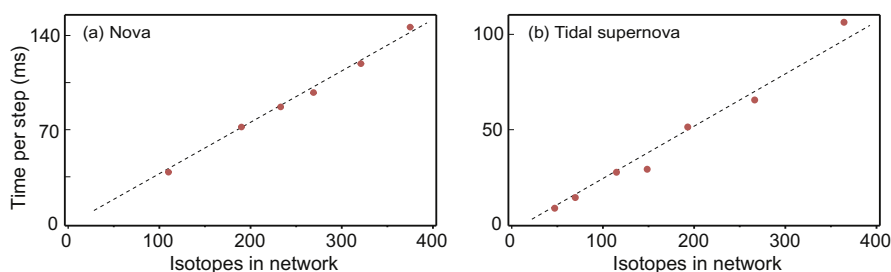
Fig. 14. Linear scaling of wall clock time per integration step with number of isotopes in the network for the explicit asymptotic approximation. (a) The nova simulations of Section 8.2. (b) The tidal supernova simulation of Section 8.3. The dashed lines are drawn only to guide the eye.

matrix operations are used, so the PE method as we have employed it here should not alter appreciably the linear scaling. We have not yet tested the PE method on very large networks, but results on networks as large as 20 species suggest that the PE overhead indeed is incremental and does not alter the scaling substantially.

11. Improving the speed of explicit codes

The preceding discussion has assumed that the relative speed of explicit and implicit methods can be compared—even if they are presently implemented with different levels of optimization—by comparing the ratio of integration timesteps required to do a particular problem with each method multiplied by a ratio of times to execute a step with each method. The assumption of this comparison is that once the matrix overhead associated with implicit methods is factored out, the rest of the work required in a timestep is similar for explicit and implicit methods. This is useful for a first estimation, and is sufficient for the primary purpose of this paper, which is to determine whether explicit methods can take stable timesteps that are even in the same ballpark as implicit methods. However, the results presented here should not be overinterpreted. Because of different levels of optimization for present implicit and explicit timestepping algorithms, and possible differences resulting from choices of integration tolerance parameters for implicit and explicit codes, the present comparisons of integration timesteps are probably uncertain by at least an order of magnitude. Thus, the essential message of the present results is that, contrary to most previous claims, explicit methods can compete favorably with implicit methods for a range of extremely stiff problems. A definitive comparison of whether implicit or explicit methods are faster for specific problems, and by how much, must await more complete development and optimization for the new explicit methods.

As has been noted, the explicit timestepping algorithm employed in this paper is serviceable but is not very optimized. Thus, it is possible that the number of integration steps required by our explicit methods has been overestimated for the examples that have been discussed. But in addition to this obvious potential for optimization, there are at least two additional reasons why we may not yet be realizing the full capability of explicit methods.

(1) For systems near equilibrium, this paper has emphasized partial equilibrium methods in conjunction with asymptotic approximations. But we have also presented evidence that quasi-steady-state (QSS) methods give results comparable to asymptotic methods, often with timesteps that can be somewhat larger. Thus, it is possible that a QSS plus partial equilibrium approximation could give even better timestepping than the asymptotic plus partial equilibrium examples shown here. This possibility has not yet been investigated.

(2) Potentially of most importance, the observation that algebraically-stabilized explicit methods may be viable for large, extremely-stiff networks changes the rules of optimization for large reaction networks. For standard implicit methods, the most effective optimizations improve the numerical linear algebra, since in large networks the bulk of the computing time for implicit methods is spent in matrix inversions. But with algebraically-stabilized explicit methods there are no matrix inversions and the majority of the time is spent in computing the rates at each step. Faster computation of reaction rates has only a small impact on the speed of an implicit code for large networks, so there has been little incentive to worry about this before. But now we see that our new explicit methods can gain even more in speed relative to implicit methods by increasing (through software or hardware) the efficiency for computing rates at each timestep. These potential speed gains are separate from, and in addition to, those accruing from the absence of matrix inversions in explicit methods that have been emphasized in preceding sections.

Let us cite a simple example illustrating this second point. We have found in a single-zone hydrodynamics calculation under Type Ia supernova conditions that using a simple rate-interpolation scheme to avoid recomputing rates for every hydrodynamical timestep permits the speed for explicit asymptotic integration for a 150-isotope network to be increased by a factor $\sim$20 in the region of strong silicon burning.

Also of interest in this context is whether the architecture of modern multicore processors with associated GPU accelerators can be exploited to compute rates more efficiently for large-scale simulations of reaction networks coupled to fluid

dynamics. We reiterate that such optimizations could be used for implicit integrations too, but they will not increase their speeds by as much because rates are a smaller part of the computing budget for implicit integration of a large network.

12. Extension of partial equilibrium to larger thermonuclear networks

Well away from equilibrium, asymptotic and QSS approximations can compete with current implicit codes even for extremely stiff systems. There are important cases where the system does not equilibrate strongly during the evolution of most physical interest (for example, those described in Sections 8.1–8.3). Even in problems where equilibrium is important overall, there will be many hydrodynamical zones and timesteps for which equilibrium plays a small role. In these situations, the asymptotic and QSS approximations alone may be adequate to solve large networks efficiently. However, to compete systematically with implicit solvers across a broad range of problems, asymptotic and QSS approximations must be augmented by PE methods to remain viable in those zones where equilibrium is important.

The examples shown here have demonstrated that asymptotic (and presumably QSS) plus partial equilibrium methods can solve stiff thermonuclear alpha networks with timestepping similar to that for implicit or semi-implicit codes, and accuracy more than sufficient to couple such an algorithm to fluid dynamics simulations. Explicit methods scale linearly and hence more favorably with network size than implicit methods, so their relative advantage grows for larger networks. Thus, for explicit methods to realize their full potential the previous partial equilibrium examples must be extended to include very stiff networks containing hundreds (or more) species. That work is ongoing and will be reported in future publications, but we note that recent encouraging results in this direction are discussed in Ref. [18].

13. Summary

Previous discussions of integrating extremely stiff reaction networks have usually concluded that such systems can be integrated only by implicit schemes, because explicit schemes are unstable for timesteps large enough to be efficient. Nevertheless, an alternative to implicit integration for stiff systems is to modify the equations to be integrated using approximate algebraic solutions to reduce the stiffness, and then to integrate the resulting equations numerically by explicit means. For example, explicit asymptotic and QSS methods have had some success in integrating moderately stiff systems in various chemical kinetics problems [1,11,12]. However, these same methods have been found wanting for the extremely stiff systems characteristic of astrophysical thermonuclear networks, both because they failed to give reliable results, and because the integration timestep was found not to be competitive with implicit methods, even if the results had been correct [1,12]. We have presented evidence strongly challenging all of this conventional wisdom.

The key to realizing this new view of explicit integration for stiff systems is the understanding that in reaction networks there are (at least) three fundamental sources of stiffness:

(1) *Negative populations*, which can evolve from an initially small positive population if an explicit timestep is too large.
(2) *Macroscopic equilibration*, where the right sides of the differential equations $dY = F = F^+ - F^-$ in Eq. (1) approach a constant.
(3) *Microscopic equilibration*, where the net flux in specific forward-reverse reaction pairs $(f_i^+ - f_i^-)$ on the right side of Eq. (1) tends to zero.

These distinctions are crucial because the algebra required to stabilize the explicit solution depends on which forms of stiffness are present in a network. The first two sources of stiffness are handled well by asymptotic and QSS methods, but microscopic equilibration can be handled efficiently using explicit methods only by employing partial equilibrium approximations.

By comparing the timesteps required to integrate problems using explicit and implicit methods, and by using an implicit backward-Euler code to estimate the relative speed for computing a timestep by explicit and implicit means in networks of various sizes, we have shown that algebraically-stabilized explicit integration can give correct results and competitive timesteps for even the stiffest of networks.

(1) For smaller networks without significant microscopic equilibration, we find systematic evidence that the explicit methods developed here can give integration speeds comparable to that for implicit integration.
(2) For some cases without much equilibration we find evidence that explicit methods might be capable of outperforming implicit ones. We have estimated that for equivalently-optimized codes the QSS method used here could be faster than the implicit method used here by factors of ∼10–15 for the nova simulation of Section 8.2, ∼10–15 for the tidal supernova alpha network of Section 8.3, and ∼5 for the tidal supernova 365-isotope network of Section 8.3 (though we have noted that these estimates are uncertain by at least an order of magnitude).
(3) For those cases where equilibration is significant the asymptotic and QSS methods give correct results, but when used alone are much too slow to compete with implicit methods. However, we presented evidence in Sections 9 and 12 that if explicit partial equilibrium methods are used to remove the stiffness associated with the approach to equilibrium, the explicit methods again exhibit speeds that are comparable to or greater than those for implicit methods.

In addition, we have argued in Section 11 that these estimates may not yet represent the best speeds for the new explicit methods. Because.

(1) our timestepping algorithm is not yet optimal,
(2) we have yet to investigate whether (as might be expected) QSS + partial equilibrium can take larger timesteps than asymptotic + partial equilibrium, and
(3) an increase (algorithmically or through hardware) in the speed for computing rates should preferentially benefit explicit methods relative to implicit ones,

It is possible that the speed of explicit methods relative to implicit ones might be increased substantially relative to the present results.

Finally, the present results owe a considerable debt to previous work [1,11–13]. However, we find that our versions of asymptotic and quasi-steady-state methods are much more accurate, and our versions of partial equilibrium methods are much faster, than those of previous authors when applied to extremely stiff astrophysical thermonuclear networks. The test cases presented here represent a mix of quite extreme conditions, with temperature changes as rapid as $\sim 10^{17}$ K/s, differences in fastest and slowest network timescales as large as 10–20 orders of magnitude, and the fraction of equilibrated reactions often changing rapidly between 0% and 100%. Hence, we conjecture that these methods may be even more useful for problems where the conditions are not quite as extreme as for astrophysical thermonuclear environments.

14. Conclusions

This paper demonstrates that algebraically-stabilized explicit integration is capable of timesteps competitive with those of implicit methods for various extremely-stiff reaction networks. Since explicit methods can execute a timestep faster than an implicit method in a large network, our results suggest that algebraically-stabilized explicit algorithms may be capable of performing as well as, or even substantially outperforming, implicit integration in a variety of moderate to extremely stiff applications. Because of the linear scaling with reaction network size for explicit methods, this fundamentally new view of explicit integration for stiff equations is particularly important for applications in fields where more realistic—and therefore larger—reaction networks are required for physical simulations. Arguably, this means almost all scientific and technical disciplines, since the sizes of reaction networks being used in simulations to this point have been dictated more often by what was feasible than by what was physical. Of particular significance is that these new explicit methods might permit coupling of more physically-realistic reaction kinetics to fluid dynamics simulations in a variety of disciplines.

Acknowledgements

Appendix A. Reaction group classification

Applying the principles discussed Section 5.4 to the reaction group classes in Table 2 gives the following partial equilibrium properties of reaction group classes for astrophysical thermonuclear networks.

Reaction group class A $(a \rightleftharpoons b)$
    Source term: $\frac{dy_a}{dt} = -k_f y_a + k_r y_b$ Constraints: $y_a + y_b \equiv c_1 = y_a^0 + y_b^0$
    Equation: $\frac{dy_a}{dt} = by_a + c$ $b = -k_f$ $c = k_r$ Solution: $y_a(t) = y_a^0 e^{bt} - \frac{c}{b}(1 - e^{bt})$
    Equil. solution: $\bar{y}_a = -\frac{c}{b} = \frac{k_r}{k_f}$ Equil. timescale: $\tau = \frac{1}{b} = \frac{1}{k_f}$
    Equil. tests: $\frac{|y_i - \bar{y}_i|}{\bar{y}_i} < \epsilon_i$ $(i = a, b)$ Equil. constraint: $\frac{y_a}{y_b} = \frac{k_r}{k_f}$
    Other variables: $y_b = c_1 - y_a$
    Progress variable: $\lambda \equiv y_a^0 - y_a$ $y_a = y_a^0 - \lambda$ $y_b = y_b^0 + \lambda$
Reaction group class B $(a + b \rightleftharpoons c)$
    Source term: $\frac{dy_a}{dt} = -k_f y_a y_b + k_r y_c$
    Constraints: $y_b - y_a \equiv c_1 = y_b^0 - y_a^0$ $y_b + y_c \equiv c_2 = y_b^0 + y_c^0$
    Equation: $\frac{dy_a}{dt} = ay_a^2 + by_a + c$ $a = -k_f$ $b = -(c_1 k_f + k_r)$ $c = k_r(c_2 - c_1)$
    Solution: Eq. (16)   Equil. solution: Eq. (17)   Equil. timescale: Eq. (20)
    Equil. tests: $\frac{|y_i - \bar{y}_i|}{\bar{y}_i} < \epsilon_i$ $(i = a, b, c)$ Equil. constraint: $\frac{y_a y_b}{y_c} = \frac{k_r}{k_f}$
    Other variables: $y_b = c_1 + y_a$ $y_c = c_2 - y_b$
    Progress variable: $\lambda \equiv y_a^0 - y_a$ $y_a = y_a^0 - \lambda$ $y_b = y_b^0 - \lambda$ $y_c = y_c^0 + \lambda$

Reaction group class C $(a + b + c \rightleftharpoons d)$

Source term: $\frac{dy_a}{dt} = -k_\mathrm{f} y_a y_b y_c + k_\mathrm{r} y_d$ Constraints: $y_a - y_b \equiv c_1 = y_a^0 - y_b^0$ $\frac{1}{3}(y_a + y_b + y_c) + y_d \equiv c_3 = \frac{1}{3}(y_a^0 + y_b^0 + y_c^0) +$ y_d^0 $y_a - y_c \equiv c_2 = y_a^0 - y_c^0$
Equation: $\frac{dy_a}{dt} = ay_a^2 + by_a + c$
$a = -k_\mathrm{f} y_a^0 + k_\mathrm{f}(c_1 + c_2)$ $b = -(k_\mathrm{f} c_1 c_2 + k_\mathrm{r})$ $c = (c_3 + \frac{1}{3} c_1 + \frac{1}{3} c_2) k_\mathrm{r}$
Solution: Eq. (16) Equil. solution: Eq. (17) Equil. timescale: Eq. (20)
Equil. tests: $\frac{|y_i - \bar{y}_i|}{\bar{y}_i} < \epsilon_i$ $(i = a, b, c, d)$ Equil. constraint: $\frac{y_a y_b y_c}{y_d} = \frac{k_\mathrm{r}}{k_\mathrm{f}}$
Other variables: $y_b = y_a - c_1$ $y_c = y_a - c_2$ $y_d = c_3 - y_a + \frac{1}{3}(c_1 + c_2)$
Progress variable: $\lambda \equiv y_a^0 - y_a$ $y_a = y_a^0 - \lambda$ $y_b = y_b^0 - \lambda$ $y_c = y_c^0 - \lambda$ $y_d = \lambda + y_d^0$

Reaction group class D $(a + b \rightleftharpoons c + d)$

Source term: $\frac{dy_a}{dt} = -k_\mathrm{f} y_a y_b + k_\mathrm{r} y_c y_d$ Constraints: $y_a - y_b \equiv c_1 = y_a^0 - y_b^0$
$y_a + y_c \equiv c_2 = y_a^0 + y_c^0$ $y_a + y_d \equiv c_3 = y_a^0 + y_d^0$
Equation: $\frac{dy_a}{dt} = ay_a^2 + by_a + c$ $a = k_\mathrm{r} - k_\mathrm{f}$ $b = -k_\mathrm{r}(c_2 + c_3) + k_\mathrm{f} c_1$ $c = k_\mathrm{r} c_2 c_3$
Solution: Eq. (16) Equil. solution: Eq. (17) Equil. timescale: Eq. (20)
Equil. tests: $\frac{|y_i - \bar{y}_i|}{\bar{y}_i} < \epsilon_i$ $(i = a, b, c, d)$ Equil. constraint: $\frac{y_a y_b}{y_c y_d} = \frac{k_\mathrm{r}}{k_\mathrm{f}}$
Other variables: $y_b = y_a - c_1$ $y_c = c_2 - y_a$ $y_d = c_3 - y_a$
Progress variable: $\lambda \equiv y_a^0 - y_a$ $y_a = y_a^0 - \lambda$ $y_b = y_b^0 - \lambda$ $y_c = y_c^0 + \lambda$ $y_d = y_d^0 + \lambda$

Reaction group class E $(a + b \rightleftharpoons c + d + e)$

Source term: $\frac{dy_a}{dt} = -k_\mathrm{f} y_a y_b + k_\mathrm{r} y_c y_d y_e$
Constraints: $y_a + \frac{1}{3}(y_c + y_d + y_e) \equiv c_1 = y_a^0 + \frac{1}{3}(y_c^0 + y_d^0 + y_e^0)$
$y_a - y_b \equiv c_2 = y_a^0 - y_b^0$ $y_c - y_d \equiv c_3 = y_c^0 - y_d^0$ $y_c - y_e \equiv c_4 = y_c^0 - y_e^0$
Equation: $\frac{dy_a}{dt} = ay_a^2 + by_a + c$
$a = (3c_1 - y_a^0) k_\mathrm{r} - k_\mathrm{f}$ $b = c_2 k_\mathrm{f} - (\alpha\beta + \alpha\gamma + \beta\gamma) k_\mathrm{r}$ $c = k_\mathrm{r} \alpha\beta\gamma$
$\alpha \equiv c_1 + \frac{1}{3}(c_3 + c_4)$ $\beta \equiv c_1 - \frac{2}{3} c_3 + \frac{1}{3} c_4$ $\gamma \equiv c_1 + \frac{1}{3} c_3 - \frac{2}{3} c_4$
Solution: Eq. (16) Equil. solution: Eq. (17) Equil. timescale: Eq. (20)
Equil. tests: $\frac{|y_i - \bar{y}_i|}{\bar{y}_i} < \epsilon_i$ $(i = a, b, c, d, e)$ Equil. constraint: $\frac{y_a y_b}{y_c y_d y_e} = \frac{k_\mathrm{r}}{k_\mathrm{f}}$
Other variables: $y_b = y_a - c_2$ $y_c = \alpha - y_a$ $y_d = \beta - y_a$ $y_e = \gamma - y_a$
Progress variable: $\lambda \equiv y_a^0 - y_a$ $y_a = y_a^0 - \lambda$ $y_b = y_b^0 - \lambda$ $y_c = y_c^0 + \lambda$
$y_d = y_d^0 + \lambda$ $y_e = y_e^0 + \lambda$

In the equilibrium test condition we have allowed the possibility of a different ϵ_i for each species i but in practice one would often choose the same small value ϵ for all i. The results presented in this paper have used $\epsilon_i = 0.01$ for all species.

References

[1] E.S. Oran, J.P. Boris, Numerical Simulation of Reactive Flow, Cambridge University Press, 2005.
[2] R.S. Arvidson, F.T. Mackenzie, M. Guidry, MAGic: a phanerozoic model for the geochemical cycling of major rock-forming components, Amer. J. Sci. 306 (2006) 135–190.
[3] W.R. Hix, B.S. Meyer, Thermonuclear kinetics in astrophysics, Nuc. Phys. A 777 (2006) 188–207.
[4] F.X. Timmes, Integration of nuclear reaction networks for stellar hydrodynamics, ApJS 124 (1999) 241–263.
[5] C.W. Gear, Numerical Initial Value Problems in Ordinary Differential Equations, Prentice Hall, 1971.
[6] J.D. Lambert, Numerical Methods for Ordinary Differential Equations, Wiley, 1991.
[7] W.H. Press, S.A. Teukolsky, W.T. Vettering, B.P. Flannery, Numerical Recipes in Fortran, Cambridge University Press, 1992.
[8] B. Fryxell, K. Olson, P. Ricker, F.X. Timmes, M. Zingale, D.Q. Lamb, P. MacNeice, R. Rosner, J. Truran, H. Tufo, FLASH: an adaptive mesh hydrodynamics code for modeling astrophysical thermonuclear flashes, ApJS 131 (2000) 273–334.
[9] M.W. Guidry, R. Budiardja, E. Feger, J.J. Billings, W.R. Hix, O.E.B. Messer, K.J. Roche, E. McMahon, M. He, Explicit integration of extremely-stiff reaction networks: asymptotic methods, arXiv:1112.4716.
[10] T.R. Young, J.P. Boris, A numerical technique for solving ordinary differential equations associated with the chemical kinetics of reactive flow problems, J. Phys. Chem. 81 (1977) 2424–2427.
[11] D.R. Mott, E.S. Oran, B. van Leer, Differential equations of reaction kinetics, J. Comp. Phys. 164 (2000) 407–428.
[12] D.R. Mott, New Quasi-Steady-State and Partial-Equilibrium Methods for Integrating Chemically Reacting Systems, doctoral thesis, University of Michigan, 1999.
[13] D. Mott, E. Oran, B. van Leer, Identifying and Imposing Partial Equilibrium in Chemically Reacting Systems, in: AIAA-2003-667, 41st Aerospace Sciences Meeting and Exhibit, Reno, Nevada, 2003.
[14] J.G. Verwer, M. van Loon, An evaluation of explicit pseudo-steady-state approximation schemes for stiff ODE systems from chemical kinetics, J. Comp. Phys. 113 (1994) 347–352.
[15] J.G. Verwer, D. Simpson, Explicit methods for stiff ODEs from atmospheric chemistry, App. Numer. Math. 18 (1995).
[16] L.O. Jay, A. Sandu, A. Porta, G.R. Carmichael, Improved quasi-steady-state-approximation methods for atmospheric chemistry integration, SIAM J. Sci. Comput. 18 (1997) 182–202.
[17] M.W. Guidry, J.A. Harris, Explicit integration of extremely-stiff reaction networks: quasi-steady-state methods, arXiv:1112.4750.
[18] M.W. Guidry, J.J. Billings, W.R. Hix, Explicit integration of extremely-stiff reaction networks: partial equilibrium methods, arXiv:1112.4738.
[19] E. Feger, Evaluating Explicit Methods for Solving Astrophysical Nuclear Reaction Networks, doctoral thesis, University of Tennessee, 2011.
[20] T. Rauscher, F.-K. Thielemann, Astrophysical reaction rates from statistical model calculations, At. Data Nuclear Data Tables 75 (2000) 1–351.
[21] W.R. Hix, F.-K. Thielemann, Computational methods for nucleosynthesis and nuclear energy generation, J. Comp. Appl. Math. 109 (1999) 321–351.
[22] Ed Anderson, LAPACK Users' Guide, 3rd Edition, SIAM 94, 1999.
[23] Iain S. Duff, A.M. Erisman, J.K. Reid, Direct Methods for Sparse Matrices, Oxford University Press, 1986.
[24] Parallel Sparse Direct Linear Solver (PARDISO) User Guide, Version 3.2, Computer Science Department, University of Basel, Switzerland (undated).; O. Schenk, K. Gartner, W. Fichtner, Efficient sparse LU factorization with left-right looking strategy on shared memory multiprocessors, BIT 40 (2000)

158–176;
O. Schenk, K. Gartner, Solving unsymmetric sparse systems of linear equations with PARDISO, J. Future Generation Comput. Syst. 20 (2004) 475–487;
O. Schenk, K. Gartner, On fast factorization pivoting methods for sparse symmetric indefinite systems, Electron. Trans. Numer. Anal. 23 (2006) 158–179.

[25] S. Parete-Koon, W.R. Hix, M.S. Smith, S. Starrfield, D.W. Bardayan, M.W. Guidry, A. Mezzacappa, Impact of a new $^{17}F(p,\gamma)$ reaction rate on nova nucleosynthesis, Astrophys. J. 598 (2003) 1239–1245.
[26] S. Parete-Koon, Reaction rate of $^{17}F(p,\gamma)^{18}Ne$ and its Implications for Nova Nucleosynthesis, Master's thesis, University of Tennessee, 2001.
[27] E. Feger, M.W. Guidry, W.R. Hix, Evaluating integration methods for astrophysical nuclear reaction networks, AP. J. Supp., submitted for publication.
[28] S. Rosswog, E. Ramirez-Ruiz, W.R. Hix, Atypical thermonuclear supernovae from tidally crushed white dwarfs, Astrophys. J. 679 (2008) 1385–1389.
[29] Tabulated at http://groups.nscl.msu.edu/jina/reaclib/db/. The JINA extensions to REACLIB are discussed in R.H. Cyburt, et al., The JINA REACLIB Database:Its Recent Updates and Impact on Type-I X-ray Bursts, Ap. J. Supp. 189 (2010) 240–252.
[30] A. Khoklov, Yet Another Stiff Solver (YASS); result quoted in Ref. [12] and algorithm described in Ref. [1].