Numer. Math. (2016) 133:623–653
DOI 10.1007/s00211-015-0760-3

Numerische Mathematik

Efficient solution of ordinary differential equations with a parametric lexicographic linear program embedded

Stuart M. Harwood[1] · Kai Höffner[1] · Paul I. Barton[1]

Received: 1 August 2013 / Revised: 16 April 2015 / Published online: 29 July 2015

Abstract This work analyzes the initial value problem in ordinary differential equations with a parametric lexicographic linear program (LP) embedded. The LP is said to be embedded since the dynamics depend on the solution of the LP, which is in turn parameterized by the dynamic states. This problem formulation finds application in dynamic flux balance analysis, which serves as a modeling framework for industrial fermentation reactions. It is shown that the problem formulation can be intractable numerically, which arises from the fact that the LP induces an effective domain that may not be open. A numerical method is developed which reformulates the system so that it is defined on an open set. The result is a system of semi-explicit index-one differential algebraic equations, which can be solved with efficient and accurate methods. It is shown that this method addresses many of the issues stemming from the original problem's intractability. The application of the method to examples of industrial fermentation processes demonstrates its effectiveness and efficiency.

Mathematics Subject Classification 34A38 · 65L05 · 90C05 · 90C99

1 Introduction

The focus of this work is the initial value problem (IVP) in ordinary differential equations (ODEs) with a parametric lexicographic linear program (LP) embedded.

This work was funded by Novartis Pharmaceuticals as part of the Novartis-MIT Center for Continuous Manufacturing.

✉ Paul I. Barton
pib@mit.edu

[1] Process Systems Engineering Laboratory, Massachusetts Institute of Technology, Cambridge, MA 02139, USA

The LP is said to be “embedded” because the vector field depends on the solution of the lexicographic LP, which is in turn parametrized by the dynamic states. See Sect. 2 for a formal problem statement. The consideration of a lexicographic LP affords a lot of modeling flexibility while simultaneously enforcing a well-defined problem. This work focuses on the situations in which this problem can be numerically intractable and when this intractability can be difficult to detect a priori. The main contribution of this work is to develop a numerical method for the solution of this problem which is accurate, efficient, and robust despite these difficulties.

The situations of interest include applications to the modeling of industrial fermentation processes. This modeling framework is known as dynamic flux balance analysis (DFBA) [14,16,18]. In its basic form, differential equations describe the evolution of the concentrations of biomass and various metabolites of interest, such as glucose or ethanol. These equations depend on the metabolism of the microbial population, which is modeled by a parametric LP. The microbes’ growth rate and uptake of resources are taken from the solution set of this LP.

One issue is that the LP may not have a singleton solution set. This means that quantities that are needed to define the dynamics of the overall system are not uniquely defined. This may lead some modelers to treat the resulting dynamic model as a differential inclusion instead. However, the ultimate goal of most research in DFBA and the motivation of this work is to obtain a numerical approximation of the solution of the dynamic problem. The idea often followed in related problems is to simulate a specifically chosen measurable selection [7,13,36]. The lexicographic LP provides a way to do exactly this by allowing the modeler to minimize or maximize various quantities in a hierarchical (or lexicographic) order over the solution set of the base LP model of the cellular metabolism. By minimizing or maximizing these quantities, a unique value for each is obtained. In essence, a specific measurable selection is chosen, and the proposed method can calculate this very efficiently. The result is that the method reduces the ambiguity of the simulation results.

Another difficulty in simulating a dynamic system with an LP embedded relates to the fact that the embedded LP can be infeasible, which could induce a closed domain of definition for the dynamic system (referred to as the “domain issue”). For typical numerical integration methods for IVPs in ODEs, this is a serious issue. Certain computations that are performed by the integration method, such as predictor steps, corrector iterations, or the calculation of Jacobian information by finite differences, require the evaluation of the dynamics at states that are near the current computed solution. When the computed solution is near the boundary of this domain of definition, these states might not be in this domain, and the result is that the numerical integrator cannot obtain the necessary information and may fail, or produce incorrect results.

Consequently, our attention goes to hybrid systems theory, where different “modes” are defined on possibly closed domains [4,12]. Typically the dynamics in those modes are trivially extended outside the domain; as in [4,12], for instance, the definition of the dynamics on an open set is given as part of the problem statement. The challenge here is defining such an extension. Thus parametric linear programming results become important [11]. This subject is concerned with the computation of the set of values that the right-hand side of the LP constraints can take and still yield a nonempty feasible set. Using results from this literature, an appropriate extension of the domain of definition of the right-hand side of the ODEs is defined. Conceptually this is similar to some parametric programming algorithms, such as those in [27].

Inspired by these results, a method is developed which redefines the system locally as index-one differential-algebraic equations (DAEs) with an open domain. The contribution of this work is the application of the parametric LP results and hybrid systems theory to the problem of ODEs with an LP embedded; this results in a powerful and implementable numerical method which is more flexible, efficient, and accurate than previous methods. Mature methods for the solution of DAEs can be used (adaptive time-stepping and error control can be used, corrector iterations defined, Jacobians are easy to obtain analytically or by finite differences). Further, the consideration of lexicographic LPs is a novel extension. This work's ability to handle the lexicographic LP in an efficient manner is a nontrivial development.

DFBA is considered in [14–16,21,33], and so these papers deal with a problem similar to the one considered here. The work in [14–16] deals with experimental validation of these models, but does not consider specific numerical issues. Meanwhile, [33] applies a differential variational inequality (DVI) formulation, and solves it with a uniform discretization in time, similarly to some time-stepping methods. This approach involves the solution of a large optimization problem (a variational inequality or mixed complementarity problem) to determine the solution trajectory all at once [1,30], and so it is very different from numerical integration methods such as the method proposed. Further, it will be seen (see Sect. 6) that ODEs with LPs embedded can be extremely stiff, which motivates the proposed developments and the ability to use numerical integration methods with adaptive time steps. The work in [21] reformulates the problem as a DAE system by replacing the embedded LP with its KKT conditions. Because of the potential for a nonunique solution set, the result is that the reformulated DAE is high-index. The subsequent need to use specialized solvers for such systems also motivates the current developments, in which an index-one DAE is obtained. As mentioned, more established numerical integration methods can be used. Finally, the aforementioned references have not explored the domain issue as it relates to DFBA, which is a significant source of numerical intractability of the ODEs with LP embedded problem. The use of a lexicographic LP distinguishes this work as well.

The rest of this work is organized as follows. Section 2 introduces notation and necessary concepts and formally states the problem. Section 3 provides motivating discussion and an example which highlights some of the difficulties inherent in the problem formulation. Section 4 considers existence and uniqueness results for the solutions of the ODE. In the context of this work, this serves as more motivation for the numerical developments. Section 5 represents the main contribution of this work, and states the proposed algorithm for solving the ODE with LP embedded problem, which includes a specific method for solving the lexicographic LP. Section 6 applies the algorithm to models of industrial fermentation processes using DFBA.

Finally, some general notation is introduced; notation specific to a section is introduced at the beginning of that section. The transposes of a vector $\mathbf{v}$ and matrix $\mathbf{M}$ are denoted $\mathbf{v}^{\mathrm{T}}$ and $\mathbf{M}^{\mathrm{T}}$, respectively. The jth component of a vector $\mathbf{v}$ is v_j and the jth column of a matrix $\mathbf{M}$ is $\mathbf{m}_j$. A vector of zeros and a vector of ones whose dimension will be implied from context will be denoted by $\mathbf{0}$ and $\mathbf{1}$, respectively. Inequalities between vectors hold componentwise. Some statements will hold at almost every $t \in I$ (i.e. except on a subset of Lebesgue measure zero), which will be abbreviated a.e. $t \in I$.

2 Problem statement and preliminaries

The formal problem statement is as follows. Let $D_t \subset \mathbb{R}$, $D_x \subset \mathbb{R}^{n_x}$ and $D_q \subset \mathbb{R}^{n_q}$ be nonempty open sets. Let $\mathbf{f} : D_t \times D_x \times D_q \to \mathbb{R}^{n_x}$, $\mathbf{b} : D_t \times D_x \to \mathbb{R}^m$, $\mathbf{A} \in \mathbb{R}^{m \times n_v}$, and $\mathbf{c}_i \in \mathbb{R}^{n_v}$ for $i \in \{1, \dots, n_q\}$ be given. First, let $\overline{\mathbb{R}} = \mathbb{R} \cup \{-\infty\} \cup \{+\infty\}$. Let $\widehat{\mathbf{q}} : \mathbb{R}^m \to \overline{\mathbb{R}}^{n_q}$ be such that

$$\begin{aligned} \widehat{q}_1(\mathbf{d}) = \inf_{\mathbf{v} \in \mathbb{R}^{n_v}} \ & \mathbf{c}_1^{\mathrm{T}} \mathbf{v} \\ \text{s.t. } & \mathbf{A}\mathbf{v} = \mathbf{d}, \\ & \mathbf{v} \geq \mathbf{0}, \end{aligned} \tag{1}$$

and for $i \in \{2, \dots, n_q\}$,

$$\begin{aligned} \widehat{q}_i(\mathbf{d}) = \inf_{\mathbf{v} \in \mathbb{R}^{n_v}} \ & \mathbf{c}_i^{\mathrm{T}} \mathbf{v} \\ \text{s.t. } & \begin{bmatrix} \mathbf{A} \\ \mathbf{c}_1^{\mathrm{T}} \\ \vdots \\ \mathbf{c}_{i-1}^{\mathrm{T}} \end{bmatrix} \mathbf{v} = \begin{bmatrix} \mathbf{d} \\ \widehat{q}_1(\mathbf{d}) \\ \vdots \\ \widehat{q}_{i-1}(\mathbf{d}) \end{bmatrix}, \\ & \mathbf{v} \geq \mathbf{0}. \end{aligned} \tag{2}$$

Subsequently, define

$$\begin{aligned} F &\equiv \left\{ \mathbf{d} \in \mathbb{R}^m : -\infty < \widehat{q}_i(\mathbf{d}) < +\infty, \ \forall i \in \{1, \dots, n_q\} \right\}, \\ K &\equiv \mathbf{b}^{-1}(F). \end{aligned} \tag{3}$$

Note that $K \subset D_t \times D_x$.

The focus of this work is an initial value problem in ODEs: given a $t_0 \in D_t$ and $\mathbf{x}_0 \in D_x$, we seek an interval $[t_0, t_f] = I \subset D_t$, and absolutely continuous function $\mathbf{x} : I \to D_x$ which satisfy

$$\dot{\mathbf{x}}(t) = \mathbf{f}\left(t, \mathbf{x}(t), \mathbf{q}(t, \mathbf{x}(t))\right), \quad \text{a.e. } t \in I, \qquad \mathbf{x}(t_0) = \mathbf{x}_0, \tag{4}$$

where $\mathbf{q} : K \to \mathbb{R}^{n_q} : (t, \mathbf{z}) \mapsto \widehat{\mathbf{q}}(\mathbf{b}(t, \mathbf{z}))$. Such an I and $\mathbf{x}$ are called a *solution* of (4).

Linear program (2) is called the ith-level LP; it is an optimization problem over the solution set of the $(i-1)$th-level LP, where the first-level LP is given by (1). Together, these LPs are called a lexicographic linear program, using the terminology from [35] (further background on lexicographic optimization is presented in Sect. 5.3).

Note that any solution of the n_qth-level LP must also be a solution of the ith-level LP, $i \in \{1, \ldots, n_q - 1\}$.

Proposition 1 establishes an important topological property of F, the domain of $\widehat{\mathbf{q}}$.

Proposition 1 *Assume F defined in (3) is nonempty. Then*

$$F = \{\mathbf{A}\mathbf{v} \in \mathbb{R}^m : \mathbf{v} \geq \mathbf{0}\},$$

and thus it is closed.

Proof Choose any $\mathbf{d} \in F$. It follows that $\widehat{q}_1(\mathbf{d})$ is finite, which implies that the first-level LP is feasible for $\mathbf{d}$; i.e. $\mathbf{A}\mathbf{v} = \mathbf{d}$ for some $\mathbf{v} \geq \mathbf{0}$. Thus $F \subset \{\mathbf{A}\mathbf{v} : \mathbf{v} \geq \mathbf{0}\}$.

Conversely, since F is nonempty, there exists $\mathbf{d}^* \in \mathbb{R}^m$ such that $\widehat{q}_i(\mathbf{d}^*)$ is finite for each i. Consequently, $\widehat{q}_1(\mathbf{d}^*) = \max\{(\mathbf{d}^*)^{\mathrm{T}}\mathbf{w} : \mathbf{A}^{\mathrm{T}}\mathbf{w} \leq \mathbf{c}_1\}$; i.e. the dual of the first-level LP is feasible and has a bounded solution. Note that the dual is feasible for any value of $\mathbf{d}$ (its feasible set is invariant). Thus, using duality results such as those in Table 4.2 of [5], $\widehat{q}_1(\mathbf{d})$ is finite for all $\mathbf{d}$ such that the first-level LP is feasible (i.e. for any $\mathbf{d} \in \{\mathbf{A}\mathbf{v} : \mathbf{v} \geq \mathbf{0}\}$).

Next, assume that $\widehat{q}_{i-1}(\mathbf{d})$ is finite for any $\mathbf{d} \in \{\mathbf{A}\mathbf{v} : \mathbf{v} \geq \mathbf{0}\}$. Since $\widehat{q}_i(\mathbf{d}^*)$ is finite, a similar argument establishes that $\widehat{q}_i(\mathbf{d})$ is finite for any $\mathbf{d} \in \{\mathbf{A}\mathbf{v} : \mathbf{v} \geq \mathbf{0}\}$. Proceeding by induction, one has that for each $i \in \{1, \ldots, n_q\}$, $\widehat{q}_i(\mathbf{d})$ is finite for any $\mathbf{d} \in \{\mathbf{A}\mathbf{v} : \mathbf{v} \geq \mathbf{0}\}$. Thus $\{\mathbf{A}\mathbf{v} : \mathbf{v} \geq \mathbf{0}\} \subset F$ and, combined with the inclusion above, equality follows. □

3 Domain issues

This section demonstrates how domain issues are manifested as numerical complications by applying the "direct" method to a simple instance of (4). To understand this from a theoretical view, note that any solution of (4) must satisfy $(t, \mathbf{x}(t)) \in K$, a.e. $t \in I$, otherwise $\mathbf{q}(t, \mathbf{x}(t))$ is undefined on a set of nonzero measure, and consequently Eq. (4) does not hold. Consequently, even though $D_t \times D_x$ is nonempty and open, the effective domain of definition of the system, K, may not be either of those.

3.1 Direct method

For this concrete example, the direct method refers to solving (4) by using a standard numerical integrator and calling an LP solver directly from the function evaluation subroutine to determine the dynamics. This approach can be made quite efficient, especially as it can rely on established commercial codes for the numerical integration and LP solution. Unfortunately, it can also be unreliable. Consider the following example:

$$\dot{\mathbf{x}}(t) = \mathbf{f}\,(\mathbf{x}(t), q(\mathbf{x}(t))) = \begin{bmatrix} 1 \\ x_2(t)q(\mathbf{x}(t)) - (x_2(t))^2 + 2x_1(t) \end{bmatrix},$$

$$
\begin{aligned}
&\mathbf{x}(0) = \mathbf{0},\\
&\qquad \text{where } q(\mathbf{z}) = \min_{v \in \mathbb{R}} v\\
&\qquad \text{s.t. } z_1^2 \le v \le z_2.
\end{aligned}
$$

The first thing to note is that the LP is feasible only if $\mathbf{z} \in K = \{\mathbf{z} : z_1^2 \le z_2\}$. Although this is a closed set, one can verify that $\mathbf{x}(t) = (t, t^2)$ is a solution; $\mathbf{x}(0) = (0, 0)$, $q(\mathbf{x}(t)) = t^2$, and $\mathbf{f}(\mathbf{x}(t), q(\mathbf{x}(t))) = (1, 2t) = \dot{\mathbf{x}}(t)$. Consider now what happens when applying an explicit Euler step. Let $\widetilde{\mathbf{x}}(t)$ be the numerical estimate of the solution at t. Then for $h > 0$ and $\widetilde{\mathbf{x}}(0) = \mathbf{x}(0)$,

$$
\begin{aligned}
\widetilde{\mathbf{x}}(0 + h) &= \widetilde{\mathbf{x}}(0) + h\mathbf{f}(\widetilde{\mathbf{x}}(0), q(\widetilde{\mathbf{x}}(0)))\\
&= \mathbf{0} + h(1, 0) = (h, 0).
\end{aligned}
$$

One sees that $\widetilde{\mathbf{x}}(h) \notin K$. Thus when attempting to evaluate $q(\widetilde{\mathbf{x}}(h))$ for the next step, one encounters an infeasible LP, and the numerical method fails.

Although explicit Euler is a very simple method, the explicit Euler step is often a part of more sophisticated integration methods; the second stage derivative of an explicit Runge–Kutta method is evaluated after taking an explicit Euler step, and the initial predictor of many linear multistep predictor-corrector methods is given by an explicit Euler step [23]. Meanwhile, numerical integration methods which do not involve an explicit Euler step will often involve an *implicit* Euler step; this includes the backwards differentiation formulas (BDF) and semi-implicit Runge–Kutta methods [23]. For the example above, an implicit method may work, but there is nothing intrinsic to an implicit method that avoids the domain issue (see Appendix B for a counterexample). In fact, implicit methods have more opportunities to fail when simulating ODEs with an LP embedded. Implicit methods typically must solve nonlinear equations by a fixed-point or Newton iteration. Since $\mathbf{f}$ and thus $\mathbf{q}$ must be evaluated at each point produced by the iteration, the sequence of iterates must be in K, which need not hold in general. Further, obtaining Jacobian information by finite differences provides another point of potential failure, as the perturbed states may not be in K.

3.2 DVI time-stepping method

Time-stepping methods refer to a class of numerical methods for solving an initial-value DVI [1,2,30,40]. The solution set of an LP is equivalent to the solution set of its KKT conditions, and the KKT conditions are a type of complementarity problem or variational inequality. Thus ODEs with an LP embedded are a special case of a DVI, and one could potentially apply a time-stepping method to (4). However, as the essential step in these methods is the solution of a system of equations with conditions that are equivalent to the embedded LP having an optimal solution, they do not differ in a meaningful way from the direct method previously mentioned. For a concrete example, see Appendix B.

4 Existence of solutions

This section presents some results for the existence and uniqueness of solutions of (4). The following theorem presents what is essentially an a posteriori check for existence. In the following λ denotes Lebesgue measure.

Theorem 1 *Suppose* $\widehat{\mathbf{q}}^E : \mathbb{R}^m \to \mathbb{R}^{n_q}$ *is an extension of* $\widehat{\mathbf{q}}$ *(i.e.* $\widehat{\mathbf{q}}^E$ *is defined on all of* $\mathbb{R}^m$ *and* $\widehat{\mathbf{q}}^E$ *restricted to* F *equals* $\widehat{\mathbf{q}}$*),* $\mathbf{b}(\cdot, \mathbf{z})$ *is measurable for all* $\mathbf{z} \in D_x$*,* $\mathbf{b}(t, \cdot)$ *is continuous for* a.e. $t \in D_t$*, and there exist an interval* $I^E = [t_0, t_f^E]$ *and absolutely continuous function* $\mathbf{x} : I^E \to D_x$ *which are a solution of the IVP*

$$\dot{\mathbf{x}}(t) = \mathbf{f}(t, \mathbf{x}(t), \widehat{\mathbf{q}}^E(\mathbf{b}(t, \mathbf{x}(t)))), \quad \text{a.e. } t \in I^E, \qquad \mathbf{x}(t_0) = \mathbf{x}_0. \tag{5}$$

Letting $S(t) = \{s \in [t_0, t] : (s, \mathbf{x}(s)) \notin K\}$*, if* $(t_0, \mathbf{x}_0) \in K$ *and*

$$t_f = \sup\{t \in I^E : \lambda(S(t)) = 0\},$$

then $I = [t_0, t_f]$ *and* $\mathbf{x}$ *restricted to* I *are a solution of IVP* (4)*. Furthermore, this is the largest interval on which* $\mathbf{x}$ *is a solution of* (4)*.*

Proof Since $\mathbf{x}$ is continuous, the composite function $\mathbf{b}_{\mathbf{x}} = \mathbf{b}(\cdot, \mathbf{x}(\cdot)) : I^E \to \mathbb{R}^m$ is measurable (see Lemma 1 in §1 of [10]). By Proposition 1, the complement of F in $\mathbb{R}^m$, F^C, is open, so $S^E = \mathbf{b}_{\mathbf{x}}^{-1}(F^C)$ is measurable. Then one has $\lambda(S(t)) = \int_{[t_0,t]} \chi_{S^E}(s) ds$, where χ_{S^E} is the indicator function of S^E. This implies that $\lambda(S(\cdot))$ is continuous and increasing.

Thus, $\lambda(S(t_f)) = 0$ and so for almost every $t \in I$, $(t, \mathbf{x}(t)) \in K$ and therefore $\mathbf{b}(t, \mathbf{x}(t)) \in F$. So $\mathbf{q}(t, \mathbf{x}(t)) = \widehat{\mathbf{q}}^E(\mathbf{b}(t, \mathbf{x}(t)))$ for almost every $t \in I$, which combined with (5) implies $\mathbf{x}$ satisfies (4) for almost every $t \in I$, and thus is a solution. The second claim follows easily; for $t' > t_f$, $\lambda(S(t')) > 0$ and so Eq. (4) cannot be satisfied for almost every $t \in [t_0, t']$. □

Refer to Eq. (5) as the "extended ODE." Note that the interval I in Theorem 1 could be degenerate, i.e. $t_0 = t_f$. This leads to a somewhat trivial solution. Ruling out this case requires something akin to the sufficient conditions for existence from viability-type results, for instance Theorem 1.2.1 of [3].

The characterization of t_f given in Theorem 1 is not in a particularly useful form. The next result alleviates this under stricter assumptions on $\mathbf{b}$.

Corollary 1 *Suppose there is a solution* $I^E = [t_0, t_f^E]$, $\mathbf{x}$ *of the extended ODE* (5)*. Let* $S(t) = \{s \in [t_0, t] : (s, \mathbf{x}(s)) \notin K\}$ *and* $t_f = \sup\{t \in I^E : \lambda(S(t)) = 0\}$*. Assume that for any* $t \in D_t$*, there exists an interval* $[t_1, t_2) \subset D_t$ *such that* $[t_1, t_2) \ni t$ *and* $\mathbf{b}$ *is continuous on* $[t_1, t_2) \times D_x$*. Then* $t_f = \inf\{t \in I^E : (t, \mathbf{x}(t)) \notin K\}$*.*

Proof For a contradiction, assume $t_f > \inf\{t \in I^E : (t, \mathbf{x}(t)) \notin K\}$, that is, there exists a $t^* \in I^E$ such that $t^* < t_f$ and $(t^*, \mathbf{x}(t^*)) \notin K$. By assumption, there is an interval $[t_1, t_2) \ni t^*$ on which $\mathbf{b}$ is continuous. Without loss of generality, assume $t_2 < t_f$. Then since $\mathbf{x}$, as a solution of the extended ODE, is continuous, $\mathbf{b}(\cdot, \mathbf{x}(\cdot))$

is continuous on $[t^*, t_2)$, and $\mathbf{b}(t^*, \mathbf{x}(t^*)) \notin F$. By Proposition 1, the complement of F, F^C, is open, so $\mathbf{b}(\cdot, \mathbf{x}(\cdot))^{-1}(F^C)$ is open in $[t^*, t_2)$ and nonempty. Thus there exists $t^{**} \in (t^*, t_2)$ such that $\mathbf{b}(t, \mathbf{x}(t)) \notin F$ for all $t \in [t^*, t^{**})$. This implies that $\lambda(S(t^{**})) > 0$. But as in the proof of Theorem 1, $\lambda(S(\cdot))$ is increasing on I^E, and so $t_f \leq t^{**}$, which contradicts $t^{**} < t_2 < t_f$.

Now, assume $t_f < \inf\{t \in I^E : (t, \mathbf{x}(t)) \notin K\}$. This implies that there exists a $t^* > t_f$ such that $(t, \mathbf{x}(t)) \in K$ for all $t < t^*$, and so $\lambda(S(t^*)) = 0$. But this contradicts the definition of t_f as a supremum. □

Corollary 1 says that, under the appropriate conditions on $\mathbf{b}$, a solution of the extended ODE ceases to be a solution of the original system (4) at the first time the solution trajectory leaves K. Intuitively this makes sense, but this intuition can lead to trouble for the numerical method as demonstrated in Sect. 3; just because one cannot find a solution of the LP at a specific step in the numerical procedure does not mean that a solution no longer exists. Care must be taken when applying Corollary 1.

Since $(t, \mathbf{z}) \mapsto \mathbf{f}(t, \mathbf{z}, \widehat{\mathbf{q}}^E(\mathbf{b}(t, \mathbf{z})))$ is defined on $D_t \times D_x$, an open set, standard existence and uniqueness results can now be applied to the extended ODE. The main concern is whether one can define an appropriate extension $\widehat{\mathbf{q}}^E$. In fact, one can define a Lipschitz continuous extension.

Proposition 2 *There exists a Lipschitz continuous function $\widehat{\mathbf{q}}^E : \mathbb{R}^m \to \mathbb{R}^{n_q}$ such that $\widehat{\mathbf{q}}^E$ restricted to F equals $\widehat{\mathbf{q}}$, the solution of the lexicographic linear program* (1)–(2).

Proof If F is empty the result is trivial. Otherwise, assume without loss of generality, that the first $k_1 = \operatorname{rank}(\mathbf{A})$ rows of $\mathbf{A}$ are linearly independent and let $\widetilde{\mathbf{A}}_1 \in \mathbb{R}^{k_1 \times n_v}$ be a matrix formed from the first k_1 rows of $\mathbf{A}$. Define $\mathbf{d}_1^E : \mathbb{R}^m \to \mathbb{R}^{k_1}$ as $\mathbf{d} \mapsto (d_1, d_2, \ldots, d_{k_1})$. Then

$$\widehat{q}_1(\mathbf{d}) = \min\{\mathbf{c}_1^{\mathrm{T}}\mathbf{v} : \widetilde{\mathbf{A}}_1\mathbf{v} = \mathbf{d}_1^E(\mathbf{d}), \mathbf{v} \geq \mathbf{0}\} \tag{6}$$

for all $\mathbf{d} \in F$. Since F is nonempty, it follows that the dual of LP (6) has a nonempty feasible set. Furthermore, by the discussion in §5.2 of [5], $\{\mathbf{p}_j \in \mathbb{R}^{k_1} : 1 \leq j \leq n_1\}$, the set of vertices of the feasible set of the dual of LP (6), is nonempty and $\widehat{q}_1(\mathbf{d}) = \max\{\mathbf{p}_j^{\mathrm{T}}\mathbf{d}_1^E(\mathbf{d}) : 1 \leq j \leq n_1\}$ for $\mathbf{d} \in F$. However, this is perfectly well-defined and Lipschitz continuous for all $\mathbf{d} \in \mathbb{R}^m$, so let

$$\widehat{q}_1^E : \mathbf{d} \mapsto \max\{\mathbf{p}_j^{\mathrm{T}}\mathbf{d}_1^E(\mathbf{d}) : 1 \leq j \leq n_1\}.$$

Then assume full row rank $\widetilde{\mathbf{A}}_i \in \mathbb{R}^{k_i \times n_v}$ and Lipschitz continuous $\widehat{q}_i^E$ and $\mathbf{d}_i^E : \mathbb{R}^m \to \mathbb{R}^{k_i}$ have been constructed such that $\widehat{q}_i^E$ restricted to F equals $\widehat{q}_i$ and $\{\mathbf{v} : \widetilde{\mathbf{A}}_i\mathbf{v} = \mathbf{d}_i^E(\mathbf{d}), \mathbf{v} \geq \mathbf{0}\}$ equals the feasible set of the ith-level LP for all $\mathbf{d} \in F$. If $\mathbf{c}_i$ and the rows of $\widetilde{\mathbf{A}}_i$ are linearly independent, let $k_{i+1} = k_i + 1$, $\widetilde{\mathbf{A}}_{i+1} = \begin{bmatrix} \widetilde{\mathbf{A}}_i \\ \mathbf{c}_i^{\mathrm{T}} \end{bmatrix}$, and $\mathbf{d}_{i+1}^E : \mathbf{d} \mapsto (\mathbf{d}_i^E(\mathbf{d}), \widehat{q}_i^E(\mathbf{d}))$; otherwise let $k_{i+1} = k_i$, $\widetilde{\mathbf{A}}_{i+1} = \widetilde{\mathbf{A}}_i$, and $\mathbf{d}_{i+1}^E : \mathbf{d} \mapsto \mathbf{d}_i^E(\mathbf{d})$. Then

$$\widehat{q}_{i+1}(\mathbf{d}) = \min\{\mathbf{c}_{i+1}^{\mathrm{T}}\mathbf{v} : \widetilde{\mathbf{A}}_{i+1}\mathbf{v} = \mathbf{d}_{i+1}^E(\mathbf{d}), \mathbf{v} \geq \mathbf{0}\} \tag{7}$$

for all $\mathbf{d} \in F$. Similarly to the induction basis, let $\{\mathbf{p}_j \in \mathbb{R}^{k_{i+1}} : 1 \leq j \leq n_{i+1}\}$ be the nonempty set of vertices of the feasible set of the dual of LP (7); then let

$$\widehat{q}_{i+1}^E : \mathbf{d} \mapsto \max\{\mathbf{p}_j^{\mathrm{T}} \mathbf{d}_{i+1}^E(\mathbf{d}) : 1 \leq j \leq n_{i+1}\}.$$

Then $\widehat{q}_{i+1}^E$ is also Lipschitz continuous, and when restricted to F it equals $\widehat{q}_{i+1}$. Proceeding by induction, one obtains the desired Lipschitz continuous extension $\widehat{\mathbf{q}}^E$. ◻

For completeness, a local existence and uniqueness result for the extended ODE is stated. Furthermore, the assumptions of the following result provide basic conditions under which the extended ODE is numerically tractable. Weakening the assumptions to allow $\mathbf{f}$ to be measurable with respect to time can be done by following results in Ch. 1 of [10].

Proposition 3 *Assume*

1. $\widehat{\mathbf{q}}^E(\mathbf{b}(t_0, \mathbf{x}_0)) \in D_q$,
2. *there exists* $t_1 > t_0$ *such that* $\mathbf{b}$ *is continuous on* $[t_0, t_1) \times D_x$ *and* $\mathbf{f}$ *is continuous on* $[t_0, t_1) \times D_x \times D_q$, *and*
3. *there exist open neighborhoods* $N_x \ni \mathbf{x}_0$ *and* $N_q \ni \widehat{\mathbf{q}}^E(\mathbf{b}(t_0, \mathbf{x}_0))$, *constants* $L_b \geq 0$, $L_f \geq 0$, *such that for all* $t \in [t_0, t_1)$, $\mathbf{z}_1, \mathbf{z}_2 \in N_x$, *and* $\mathbf{p}_1, \mathbf{p}_2 \in N_q$

$$\|\mathbf{b}(t, \mathbf{z}_1) - \mathbf{b}(t, \mathbf{z}_2)\| \leq L_b \|\mathbf{z}_1 - \mathbf{z}_2\|,$$
$$\|\mathbf{f}(t, \mathbf{z}_1, \mathbf{p}_1) - \mathbf{f}(t, \mathbf{z}_2, \mathbf{p}_2)\| \leq L_f(\|\mathbf{z}_1 - \mathbf{z}_2\| + \|\mathbf{p}_1 - \mathbf{p}_2\|).$$

Then a unique solution of the IVP (5) *exists.*

Proof By Proposition 2, one can assume $\widehat{\mathbf{q}}^E$ is Lipschitz continuous with constant L_q, so $\mathbf{q}^E = \widehat{\mathbf{q}}^E \circ \mathbf{b}$ is continuous on $[t_0, t_1) \times D_x$ and satisfies

$$\|\mathbf{q}^E(t, \mathbf{z}_1) - \mathbf{q}^E(t, \mathbf{z}_2)\| \leq L_q L_b \|\mathbf{z}_1 - \mathbf{z}_2\|.$$

Since $\mathbf{q}^E$ is continuous, one can assume without loss of generality that $\mathbf{q}^E(t, \mathbf{z}) \in N_q$ for all $(t, \mathbf{z}) \in [t_0, t_1) \times N_x$. Thus,

$$\|\mathbf{f}(t, \mathbf{z}_1, \mathbf{q}^E(t, \mathbf{z}_1)) - \mathbf{f}(t, \mathbf{z}_2, \mathbf{q}^E(t, \mathbf{z}_2))\| \leq L_f(1 + L_q L_b)\|\mathbf{z}_1 - \mathbf{z}_2\|.$$

Therefore one can apply Thm. 2.3 of Ch. II of [26] to the mapping $(t, \mathbf{z}) \mapsto \mathbf{f}(t, \mathbf{z}, \mathbf{q}^E(t, \mathbf{z}))$ and conclude that there exists a $t_f^E > t_0$ and continuous function $\mathbf{x}$ on $[t_0, t_f^E]$ which are a solution of (5). ◻

5 Numerical developments

This section discusses the numerical method that has been developed for the efficient and reliable integration of ODEs with LP embedded. First, notation specific to this section and background from linear programming are introduced in Sect. 5.1. Then the overall numerical integration routine is introduced in Sect. 5.2. This method depends on a specific way to solve the lexicographic LP (1)–(2), which is described in Sect. 5.3.

5.1 Notation and background

Consider a vector $\mathbf{v} \in \mathbb{R}^n$ and a matrix $\mathbf{M} \in \mathbb{R}^{p \times n}$. For an index set $J = \{j_1, \ldots, j_{n_J}\} \subset \{1, \ldots, n\}$, let $\mathbf{v}_J = (v_{j_1}, \ldots, v_{j_{n_J}})$ and similarly $\mathbf{M}_J = \left[\mathbf{m}_{j_1} \cdots \mathbf{m}_{j_{n_J}}\right]$. Similar notation applies to vectors and matrices that already have a subscript. For instance, $c_{i,j}$ is the jth component of the vector $\mathbf{c}_i$, and for some index set $J \subset \{1, \ldots, n_v\}$, $\mathbf{c}_{i,J}$ is the vector formed from the components of $\mathbf{c}_i$ corresponding to J. In Algorithm 2 and Theorem 2 matrices $\mathbf{A}_i$, $i \in \{1, \ldots, n_q\}$, will be constructed. It will be useful to think of their columns as indexed by some set P_i. Thus, for $j \in P_i$ and $J \subset P_i$, the jth column of $\mathbf{A}_i$ is denoted $\mathbf{a}_{i,j}$, and $\mathbf{A}_{i,J}$ is the matrix formed from the columns of $\mathbf{A}_i$ corresponding to J. The cardinality of a set J is $\operatorname{card}(J)$.

The following linear programming background will be helpful, which draws freely from the first four chapters of [5]. Consider the first-level LP as a prototype for standard-form LPs parameterized by the right-hand side of the constraints:

$$\widehat{q}_1(\mathbf{d}) = \inf\{\mathbf{c}_1^{\mathrm{T}}\mathbf{v} : \mathbf{v} \in \mathbb{R}^{n_v}, \mathbf{A}\mathbf{v} = \mathbf{d}, \mathbf{v} \geq \mathbf{0}\}. \tag{8}$$

The following assumption will hold in this and subsequent sections. It is a standard assumption of the simplex method, upon which the proposed numerical developments are based.

Assumption 1 *The matrix* $\mathbf{A}$ *is full row rank.*

The concept of a basis is introduced. A basis B is a subset of $\{1, \ldots, n_v\}$ with $m = \operatorname{card}(B)$. An optimal basis is one which satisfies

$$\mathbf{A}_B^{-1}\mathbf{d} \geq \mathbf{0}, \tag{9}$$

$$\mathbf{c}_1^{\mathrm{T}} - \mathbf{c}_{1,B}^{\mathrm{T}}\mathbf{A}_B^{-1}\mathbf{A} \geq \mathbf{0}^{\mathrm{T}}. \tag{10}$$

A basis which satisfies (9) is primal feasible, while one that satisfies (10) is dual feasible. Thus, a basis is optimal if and only if it is primal and dual feasible. The invertible matrix $\mathbf{A}_B$ is the corresponding basis matrix. A basis also serves to describe a vector $\mathbf{v} \in \mathbb{R}^{n_v}$; the components of the vector corresponding to B, $\mathbf{v}_B$, are given by $\mathbf{v}_B = \mathbf{A}_B^{-1}\mathbf{d}$, and the rest are zero, i.e. $v_j = 0$, $j \notin B$. If the basis B is optimal, then the vector $\mathbf{v}$ which it describes is in the optimal solution set of the first-level LP. Thus, $\widehat{q}_1(\mathbf{d}) = \mathbf{c}_1^{\mathrm{T}}\mathbf{v} = \mathbf{c}_{1,B}^{\mathrm{T}}\mathbf{v}_B$. The variables $\mathbf{v}_B$ are called the basic variables. The vector $\mathbf{c}_1^{\mathrm{T}} - \mathbf{c}_{1,B}^{\mathrm{T}}\mathbf{A}_B^{-1}\mathbf{A}$ is the vector of reduced costs. It is clear that perturbations in $\mathbf{d}$ do not affect dual feasibility of a basis. Thus, a basis is optimal for all $\mathbf{d}$ such that the basic variables are nonnegative.

5.2 Solution algorithm

Theorem 1, Corollary 1, and Proposition 3 indicate how one should approach calculating a solution of IVP (4): solve the extended ODE (5) and detect the earliest time that the solution trajectory leaves K, indicated by the infeasibility of the embedded LP at a point on the solution trajectory. In general terms, this is the approach taken in the following numerical method. Under the assumptions of Proposition 3, broad classes of numerical integration methods are convergent for the extended ODE (5), including linear multistep and Runge–Kutta methods [23]. However, there is still the issue that one needs to detect the earliest time that the solution trajectory leaves K accurately and reliably. As indicated by the examples in Sect. 3, one cannot merely rely on detecting an infeasible embedded LP, as this could occur during a corrector iteration or be due to poor integration error control. The following method addresses these issues.

The essence of the method is easily understood when $n_q = 1$, in which case the dynamics only depend on the optimal objective value of a single LP parameterized by its right-hand side. If one solves the embedded LP at the initial conditions with any method which finds an optimal basis B, then for as long as B is optimal, one can obtain the optimal basic variables by solving the system $\mathbf{A}_B\mathbf{u}_B(t) = \mathbf{b}(t, \mathbf{x}(t))$ for $\mathbf{u}_B(t)$, from which one obtains $q_1(t, \mathbf{x}(t)) = \mathbf{c}_{1,B}^{\mathrm{T}}\mathbf{u}_B(t)$. Meanwhile, B is optimal for as long as the basic variables are nonnegative, i.e. $\mathbf{u}_B(t) \geq \mathbf{0}$. Consequently, the general idea is to reformulate the system as DAEs, where the basic variables $\mathbf{u}_B$ have been added as algebraic variables, and employ event detection to detect when the value of a basic variable crosses zero. Once a basic variable crosses zero, a new optimal basis is found by re-solving the LP, and the procedure is repeated.

For the time being suppose that a δ-optimal basis B is acceptable; that is to say that $\mathbf{A}_B^{-1}\mathbf{b}(t, \mathbf{x}(t)) > -\delta\mathbf{1}$. To guarantee the detection of when B ceases to be δ-optimal, one needs to use a feasibility tolerance $\epsilon < \delta$ when solving the embedded LP. Then, the initial values of the basic variables satisfy

$$\mathbf{u}_B(t_0) \geq -\epsilon\mathbf{1} > -\delta\mathbf{1}.$$

Consequently, $\mathbf{u}_B(t_0)+\delta\mathbf{1}$ is strictly positive. If B ceases to be δ-optimal, then for some index j, the value $u_j(t) + \delta$ will cross zero, which can be detected quite accurately with event detection algorithms [31]. The following DAEs, while $\mathbf{u}_B(t) > -\delta\mathbf{1}$, are integrated numerically:

$$\begin{aligned} \dot{\mathbf{x}}(t) - \mathbf{f}(t, \mathbf{x}(t), \mathbf{c}_{1,B}^{\mathrm{T}}\mathbf{u}_B(t)) &= \mathbf{0}, \\ \mathbf{A}_B\mathbf{u}_B(t) - \mathbf{b}(t, \mathbf{x}(t)) &= \mathbf{0}. \end{aligned}$$

Since $\mathbf{A}_B$ is nonsingular, it is clear from inspection that this is a semi-explicit index-one system of DAEs, and amenable to many numerical integration methods.

Of course, ϵ is a small, but positive, number, and so δ must be as well. Consequently, one has to ask whether it actually is acceptable for the basis B to be merely δ-optimal. Since the goal is to calculate a solution of the extended ODE, one needs to ensure that for a δ-optimal basis B, $\widehat{q}_1^B(\mathbf{d}) = \mathbf{c}_{1,B}^{\mathrm{T}}\mathbf{A}_B^{-1}\mathbf{d}$ is an accurate approximation of

$\widehat{q}_1^E(\mathbf{d})$. Indeed it is. For a dual feasible basis B, let $F_B = \{\mathbf{d} \in \mathbb{R}^m : \mathbf{A}_B^{-1}\mathbf{d} \geq \mathbf{0}\}$, thus F_B is the subset of F on which B is primal feasible and so also optimal. Let $F_{B,\delta} = \{\mathbf{d} \in \mathbb{R}^m : \mathbf{A}_B^{-1}\mathbf{d} \geq -\delta\mathbf{1}\}$, thus $F_{B,\delta}$ is the set on which B is δ-optimal. Now assume $\mathbf{d} \in F_{B,\delta}$ and let $\mathbf{v} = \mathbf{A}_B^{-1}\mathbf{d}$. Construct $\widetilde{\mathbf{v}}$ such that $\widetilde{v}_i = \max\{v_i, 0\}$, thus $\widetilde{\mathbf{v}} \geq \mathbf{0}$. Let $\widetilde{\mathbf{d}} = \mathbf{A}_B\widetilde{\mathbf{v}} \in F_B$. Note that $\|\mathbf{v} - \widetilde{\mathbf{v}}\|_\infty \leq \delta$, thus $\|\mathbf{d} - \widetilde{\mathbf{d}}\|_\infty \leq \|\mathbf{A}_B\|_\infty\delta$. Since $\widehat{q}_1^B = \widehat{q}_1^E$ on F_B, $\widehat{q}_1^B(\widetilde{\mathbf{d}}) = \widehat{q}_1^E(\widetilde{\mathbf{d}})$. Consequently,

$$\begin{aligned}
|\widehat{q}_1^B(\mathbf{d}) - \widehat{q}_1^E(\mathbf{d})| &\leq |\widehat{q}_1^B(\mathbf{d}) - \widehat{q}_1^B(\widetilde{\mathbf{d}})| + |\widehat{q}_1^B(\widetilde{\mathbf{d}}) - \widehat{q}_1^E(\mathbf{d})| \\
&= |\widehat{q}_1^B(\mathbf{d}) - \widehat{q}_1^B(\widetilde{\mathbf{d}})| + |\widehat{q}_1^E(\widetilde{\mathbf{d}}) - \widehat{q}_1^E(\mathbf{d})| \\
&\leq \|\mathbf{c}_{1,B}^{\mathrm{T}}\mathbf{A}_B^{-1}\|_2\|\mathbf{d} - \widetilde{\mathbf{d}}\|_2 + L_q\|\mathbf{d} - \widetilde{\mathbf{d}}\|_2 \\
&\leq M\delta,
\end{aligned}$$

where the second to last inequality follows from the Lipschitz continuity of $\widehat{q}_1^E$. Note that M is finite and can be chosen so that the inequality holds for any choice of B, since there are a finite number of dual feasible bases. Thus, the error in approximating $\widehat{q}_1^E$ using a δ-optimal basis must go to zero as δ goes to zero.

The failure to find a δ-optimal basis at a particular value of $\mathbf{b}(t, \mathbf{x}(t))$ simply implies that $(t, \mathbf{x}(t)) \notin K$. If a δ-optimal basis does not exist, then certainly an optimal basis does not exist, which means that $\mathbf{b}(t, \mathbf{x}(t)) \notin F$ implying $(t, \mathbf{x}(t)) \notin K$, and so by Corollary 1, the calculated solution is no longer a solution of (4). However, since the test of whether $\mathbf{b}(t, \mathbf{x}(t)) \notin F$ is only performed as part of the determination of a new optimal basis, after the old one has stopped being δ-optimal, this is a much more reliable indication that the solution cannot be continued.

To generalize this method to the case $n_q > 1$ the overall structure remains unchanged. This is because it is possible to find a basis B which is optimal for the first-level LP and which describes a point which is in the optimal solution set of the ith-level LP, for *all* i. Then $\widehat{\mathbf{q}}(\mathbf{d}) = (\mathbf{c}_{1,B}^{\mathrm{T}}\mathbf{A}_B^{-1}\mathbf{d}, \ldots, \mathbf{c}_{n_q,B}^{\mathrm{T}}\mathbf{A}_B^{-1}\mathbf{d})$ for all $\mathbf{d}$ such that B is optimal for the first-level LP. This idea is proved in Theorem 2 and the method for determining the appropriate basis is summarized in Algorithm 2, both presented in Sect. 5.3.

The numerical method in the general case is summarized in Algorithm 1. An empty basis set returned by Algorithm 2 serves as a flag that $\mathbf{b}(t, \mathbf{x}(t)) \notin F$ and that the solution cannot be continued. The convergence of Algorithm 1 (as δ and step size tend to zero) is guaranteed if the numerical method used to integrate the DAE system (11) is convergent for the extended ODE (5), which, as mentioned earlier, includes broad classes under the assumptions of Proposition 3. This follows from simple arguments for the convergence of methods for semi-explicit index-one DAEs; see for instance §3.2.1 of [6]. Overall, Algorithm 1 produces an approximation of the solution of the extended ODE, and gives a reliable and accurate indication of when this solution is no longer a solution of the original system (4).

An implementation of Algorithm 1 has been coded incorporating DAEPACK [39] component DSL48E for the numerical integration of the DAE and event detection. DSL48E uses a BDF method and the sparse unstructured linear algebra code MA48 [9], and so is appropriate for the numerical integration of stiff systems; these features

Algorithm 1 Overall solution method for the IVP (4)

Require: $\delta > \epsilon > 0$, $t_f > t_0$
  $\widetilde{t} \leftarrow t_0$, $\widetilde{\mathbf{x}} \leftarrow \mathbf{x}_0$
  loop
    $B \leftarrow B^*(\mathbf{b}(\widetilde{t}, \widetilde{\mathbf{x}}), \epsilon)$ (See Algorithm 2)
    if $B = \varnothing$ **then**
      Terminate.
    end if
    Solve $\mathbf{A}_B \widetilde{\mathbf{u}}_B = \mathbf{b}(\widetilde{t}, \widetilde{\mathbf{x}})$ for $\widetilde{\mathbf{u}}_B$.
    Set $\mathbf{q}^B : \mathbf{u} \mapsto (\mathbf{c}_{1,B}^{\mathrm{T}}\mathbf{u}, \ldots, \mathbf{c}_{n_q,B}^{\mathrm{T}}\mathbf{u})$.
    while $\widetilde{\mathbf{u}}_B > -\delta\mathbf{1}$ **do**
      Update $\left(\widetilde{t}, \widetilde{\mathbf{x}}, \widetilde{\mathbf{u}}_B\right)$ by integrating the following DAE system with an appropriate method:

$$\begin{aligned} \dot{\mathbf{x}}(t) - \mathbf{f}(t, \mathbf{x}(t), \mathbf{q}^B(\mathbf{u}_B(t))) &= \mathbf{0}, \\ \mathbf{A}_B \mathbf{u}_B(t) - \mathbf{b}(t, \mathbf{x}(t)) &= \mathbf{0}. \end{aligned} \tag{11}$$

      if $\widetilde{t} \geq t_f$ **then**
        Terminate.
      end if
    end while
  end loop

will be indispensable in the solution of DFBA models in Sect. 6. Meanwhile, the event detection algorithm is an accurate and efficient method developed in [31]. A code employing CPLEX implements Algorithm 2. This implementation of the algorithms has been named DSL48LPR.

5.3 Lexicographic optimization

An inefficient way to try to generalize the basic idea behind Algorithm 1 to $n_q > 1$ would be to calculate an optimal basis for each level LP, disregarding the connections between the levels.

However, by exploiting the relationship between the individual levels in the lexicographic LP, it in fact suffices to determine a *single* optimal basis for the first-level LP (1) to calculate some element of the solution set of the ith-level LP for each i. Theorem 2 formalizes this and its proof provides a constructive method of finding the appropriate basis. The construction is summarized in Algorithm 2.

The benefit of Algorithm 2 is that it allows one to use standard primal simplex. That is, any pivot selection rules can be used, and so one can rely on a commercial implementation of primal simplex to implement Algorithm 2, and then degeneracy and cycling are not a concern. Modifications of the simplex algorithm ("lexicographic simplex") have been presented in Ch. 3 of [19], §10.5 of [38], and [20,22,32] to solve lexicographic LPs. These methods are similar in effect to Algorithm 2. In contrast, these methods either do not consider the parametric results needed here, require specific pivot selection rules, or do not consider degeneracy or cycling.

Theorem 2 *Assume that* $\mathbf{d} \in F$. *Then there exists a basis* B_1^* *that is optimal for the first-level LP* (1) *and*

$$\widehat{\mathbf{q}}(\mathbf{d}) = \left(\mathbf{c}_{1,B_1^*}^{\mathrm{T}} \mathbf{A}_{B_1^*}^{-1} \mathbf{d}, \ldots, \mathbf{c}_{n_q,B_1^*}^{\mathrm{T}} \mathbf{A}_{B_1^*}^{-1} \mathbf{d}\right). \tag{12}$$

Further, this relation holds for all $\mathbf{d}$ *such that* B_1^* *is optimal for the first-level LP.*

Proof Existence and construction of the appropriate basis proceed by induction; again, the construction is summarized in Algorithm 2. At each induction step a special "projected" LP is constructed and optimized. The reason behind considering this projected LP is that we can draw conclusions about the pivots taken when optimizing it with primal simplex. This allows us to argue about the form of the optimal basis. It is suggested that the reader study the results in Appendix A before proceeding.

First introduce some notation. $\mathbf{A}_i$ denotes a specifically constructed matrix. For some index set J, the matrix $\mathbf{A}_i^J$ is the matrix equaling $\mathbf{A}_i$ with those columns corresponding to J set to $\mathbf{0}$.

Fix $\mathbf{d} \in F$ to the value of interest. For an induction basis, let B_1 be any optimal basis for the first-level LP (1), $n_1 = n_v, m_1 = m$, $P_1 = \{1, \ldots, n_v\}$, $N_1 = \varnothing$, $\mathbf{A}_1 = \mathbf{A}$ and $\mathbf{d}_1(\mathbf{d}) = \mathbf{d}$. An optimal tableau for the first-level LP is

$$\mathbf{A}_{1,B_1}^{-1} \left[\mathbf{d}_1 \ \mathbf{A}_1\right] = \left[\mathbf{A}_{B_1}^{-1}\mathbf{d} \ \mathbf{A}_{B_1}^{-1}\left(\mathbf{A}_{P_1}^{N_1}\right)\right].$$

For the ith induction step assume the following:

1. Assume for $k \in \{2, \ldots, i\}$, $n_{k-1} \geq n_k$, $m_{k-1} \leq m_k$, $N_{k-1} \subset N_k$, and for $k \in \{1, \ldots, i\}$, $P_k = \{1, \ldots, n_k\}$, $N_k \subset P_k$, $\mathbf{A}_k \in \mathbb{R}^{m_k \times n_k}$ and $\mathbf{d}_k : F \to \mathbb{R}^{m_k}$. Consider the kth "projected" LPs, for $k \in \{1, \ldots, i\}$
$$\begin{aligned} q_k^P(\mathbf{d}) = \min_{\mathbf{v} \in \mathbb{R}^{n_k}} \ & \mathbf{c}_{k,P_k}^{\mathrm{T}} \mathbf{v} \\ \text{s.t.} \ & \mathbf{A}_k \mathbf{v} = \mathbf{d}_k(\mathbf{d}), \\ & \mathbf{v} \geq \mathbf{0}. \end{aligned} \tag{13}$$
2. Assume that the ith-level LP (2) is equivalent to the ith projected LP in the sense of Definition 1 in Appendix A.
3. Assume the bases B_1, and for $k \in \{2, \ldots, i\}$, $B_k = N_k \cup B_1$ are optimal for the first-level and kth projected LPs, respectively. Also assume that for $k \in \{1, \ldots, i-1\}$, $c_{k,j} - \mathbf{c}_{k,B_k}^{\mathrm{T}} \mathbf{A}_{k,B_k}^{-1} \mathbf{a}_{k,j} > 0$ for each $j \in (P_k \backslash P_{k+1}) \cup (N_{k+1} \backslash N_k)$, and $c_{k,j} - \mathbf{c}_{k,B_k}^{\mathrm{T}} \mathbf{A}_{k,B_k}^{-1} \mathbf{a}_{k,j} = 0$ for each $j \in P_i \backslash B_i$.
4. Assume that the tableau for the ith projected LP resulting from the basis B_i is
$$\mathbf{A}_{i,B_i}^{-1} [\mathbf{d}_i(\mathbf{d}) \quad \mathbf{A}_i] = \begin{bmatrix} \mathbf{0} & \mathbf{E}_i \\ \mathbf{A}_{B_1}^{-1}\mathbf{d} & \mathbf{A}_{B_1}^{-1}\left(\mathbf{A}_{P_i}^{N_i}\right) \end{bmatrix},$$

Algorithm 2 Method for determining optimal basis for lexicographic LP (1)-(2)

Require: $\mathbf{d} \in \mathbb{R}^m$, $\epsilon > 0$
  $P_1 \leftarrow \{1, \ldots, n_v\}$
  $n_1 \leftarrow n_v$, $N_1 \leftarrow \varnothing$
  $\mathbf{A}_1 \leftarrow \mathbf{A}$, $\mathbf{d}_1 \leftarrow \mathbf{d}$
  Solve first-level LP with absolute feasibility tolerance ϵ:

$$q_1^* = \inf_{\mathbf{v} \in \mathbb{R}^n} \mathbf{c}_1^{\mathrm{T}} \mathbf{v} \quad \text{s.t. } \mathbf{A}\mathbf{v} = \mathbf{d}, \quad \mathbf{v} \geq \mathbf{0}.$$

  if $-\infty < q_1^* < +\infty$ **then**
    Determine optimal basis B_1 for first-level LP.
  else
    $B^*(\mathbf{d}, \epsilon) \leftarrow \varnothing$
    Terminate.
  end if
  $i \leftarrow 1$
  while $i < n_q$ **do**
    if $c_{i,j} - \mathbf{c}_{i,B_i}^{\mathrm{T}} \mathbf{A}_{i,B_i}^{-1} \mathbf{a}_{i,j} > 0, \forall j \in P_i \backslash B_i$ **then**
      $B^*(\mathbf{d}, \epsilon) \leftarrow B_1$
      Terminate.
    end if
    if $\mathbf{c}_{i,P_i}^{\mathrm{T}} - \mathbf{c}_{i,B_i}^{\mathrm{T}} \mathbf{A}_{i,B_i}^{-1} \mathbf{A}_i = \mathbf{0}^{\mathrm{T}}$ **then**
      $P_{i+1} \leftarrow P_i$
      $n_{i+1} \leftarrow n_i$, $N_{i+1} \leftarrow N_i$
      $\mathbf{A}_{i+1} \leftarrow \mathbf{A}_i$, $\mathbf{d}_{i+1} \leftarrow \mathbf{d}_i$
    else
      Choose $j \in P_i$ such that $c_{i,j} - \mathbf{c}_{i,B_i}^{\mathrm{T}} \mathbf{A}_{i,B_i}^{-1} \mathbf{a}_{i,j} > 0$.
      $P_{i+1} = \left\{ k \in P_i : c_{i,k} - \mathbf{c}_{i,B_i}^{\mathrm{T}} \mathbf{A}_{i,B_i}^{-1} \mathbf{a}_{i,k} = 0 \right\} \cup \{j\}$
      $n_{i+1} \leftarrow \mathrm{card}(P_{i+1})$, $N_{i+1} \leftarrow N_i \cup \{j\}$
      $\mathbf{A}_{i+1} \leftarrow \begin{bmatrix} \mathbf{c}_{i,P_{i+1}}^{\mathrm{T}} \\ \mathbf{A}_{i,P_{i+1}} \end{bmatrix}$, $\mathbf{d}_{i+1} \leftarrow \begin{bmatrix} q_i^* \\ \mathbf{d}_i \end{bmatrix}$
    end if
    Solve $(i+1)^{th}$ projected LP with primal simplex using initial basis $B_1 \cup N_{i+1}$ and absolute feasibility tolerance ϵ:

$$q_{i+1}^* = \inf_{\mathbf{v} \in \mathbb{R}^{n_{i+1}}} \mathbf{c}_{i+1,P_{i+1}}^{\mathrm{T}} \mathbf{v} \quad \text{s.t. } \mathbf{A}_{i+1}\mathbf{v} = \mathbf{d}_{i+1}, \quad \mathbf{v} \geq \mathbf{0}.$$

    if $-\infty < q_{i+1}^* < +\infty$ **then**
      For $(i+1)^{th}$ projected LP, optimal basis is $B_{i+1} = \widetilde{B}_1 \cup N_{i+1}$.
      $B_1 \leftarrow \widetilde{B}_1$
    else
      $B^*(\mathbf{d}, \epsilon) \leftarrow \varnothing$
      Terminate.
    end if
    $i \leftarrow i + 1$
    $B^*(\mathbf{d}, \epsilon) \leftarrow B_1$
  end while
  return $B^*(\mathbf{d}, \epsilon)$

where $\mathbf{E}_i$ is a $(m_i - m_1) \times n_i$ matrix constructed from the rows of the $n_i \times n_i$ identity matrix that correspond to elements of N_i. Recall that the left-most column of the above tableau is typically called the "zeroth" column.

There are three cases when constructing the next LP. In the first case, consider the reduced costs for the ith projected LP determined from the basis B_i from assumption 3. If each reduced cost corresponding to a nonbasic variable is positive (i.e. $\forall j \in P_i \backslash B_i$, $c_{i,j} - \mathbf{c}^{\mathrm{T}}_{i,B_i}\mathbf{A}^{-1}_{i,B_i}\mathbf{a}_{i,j} > 0$), then the point described by the basis B_i is the unique optimal solution point for the ith projected LP [5]. By assumption of equivalence, the solution set of the ith-level LP is also a singleton; let this point be $\mathbf{v}^* \in \mathbb{R}^{n_v}$. Combined with assumption 4, the only nonzero components of $\mathbf{v}^*$ are those corresponding to B_1, so we have $\mathbf{c}_1^{\mathrm{T}}\mathbf{v}^* = \mathbf{c}^{\mathrm{T}}_{1,B_1}\mathbf{A}^{-1}_{B_1}\mathbf{d}$. Of course, by the nature of the lexicographic LP, $\mathbf{v}^*$ must be an optimal solution point of the kth-level LP, for all $k \in \{1, \ldots, n_q\}$, and so letting $B_1^* = B_1$ we have that Eq. (12) holds.

For the other two cases, a higher-level LP must be considered. Our aim is to construct the $(i+1)$th projected LP

$$\begin{aligned} q^P_{i+1}(\mathbf{d}) = \min_{\mathbf{v}\in\mathbb{R}^{n_{i+1}}} \ & \mathbf{c}^{\mathrm{T}}_{i+1,P_{i+1}}\mathbf{v} \\ \text{s.t.} \ & \mathbf{A}_{i+1}\mathbf{v} = \mathbf{d}_{i+1}(\mathbf{d}), \\ & \mathbf{v} \geq \mathbf{0}. \end{aligned} \tag{14}$$

In the second case, if $\mathbf{c}^{\mathrm{T}}_{i,P_i} - \mathbf{c}^{\mathrm{T}}_{i,B_i}\mathbf{A}^{-1}_{i,B_i}\mathbf{A}_i = \mathbf{0}^{\mathrm{T}}$, then $\mathbf{c}^{\mathrm{T}}_{i,P_i}$ and the rows of $\mathbf{A}_i$ are linearly dependent, and so the constraint $\mathbf{c}^{\mathrm{T}}_{i,P_i}\mathbf{v} = q_i^P(\mathbf{d})$ is redundant (it is satisfied everywhere in the feasible set of the ith projected LP). Let $n_{i+1} = n_i$, $m_{i+1} = m_i$, $P_{i+1} = P_i$, $N_{i+1} = N_i$, $\mathbf{A}_{i+1} = \mathbf{A}_i$ and $\mathbf{d}_{i+1} = \mathbf{d}_i$. The basis $B_{i+1} = B_i$ is primal feasible for the $(i+1)$th projected LP. To help establish that induction assumption 3 will hold for the $(i+1)$th step, note that we trivially have $c_{i,j} - \mathbf{c}^{\mathrm{T}}_{i,B_i}\mathbf{A}^{-1}_{i,B_i}\mathbf{a}_{i,j} > 0$ for each $j \in (P_i \backslash P_{i+1}) \cup (N_{i+1} \backslash N_i)$, and $c_{i,j} - \mathbf{c}^{\mathrm{T}}_{i,B_i}\mathbf{A}^{-1}_{i,B_i}\mathbf{a}_{i,j} = 0$ for each $j \in P_{i+1} \backslash B_{i+1}$. The resulting tableau is the same form as in assumption 4. It is clear that the feasible set of the $(i+1)$th projected LP is the solution set of the ith projected LP; by the induction assumption of equivalence and Lemma 1 in Appendix A, we have that the $(i+1)$th projected LP is equivalent to the $(i+1)$th-level LP.

In the third case, if there is a $j \in P_i$ such that $c_{i,j} - \mathbf{c}^{\mathrm{T}}_{i,B_i}\mathbf{A}^{-1}_{i,B_i}\mathbf{a}_{i,j} > 0$, then let

$$P_{i+1} = \left\{k \in P_i : c_{i,k} - \mathbf{c}^{\mathrm{T}}_{i,B_i}\mathbf{A}^{-1}_{i,B_i}\mathbf{a}_{i,k} = 0\right\} \cup \{j\}.$$

Let n_{i+1} be the number of elements in P_{i+1} and assume without loss of generality that $P_{i+1} = \{1, \ldots, n_{i+1}\}$ (the variables could be re-ordered as necessary). Let $m_{i+1} = m_i + 1$, $N_{i+1} = \{j\} \cup N_i$ and $B_{i+1} = N_{i+1} \cup B_1$. Note that $B_{i+1} = \{j\} \cup N_i \cup B_1$, and since $B_i = N_i \cup B_1$, we have $B_{i+1} = \{j\} \cup B_i$. From Lemma 2 in Appendix A, we have that B_{i+1} is primal feasible for the $(i+1)$th projected LP. Since the basic variables of the ith projected LP have corresponding reduced costs that are zero, from the definition of P_{i+1} we have $B_{i+1} \subset P_{i+1}$ so this is a well-defined basis. To help establish that induction assumption 3 will hold for the $(i+1)$th step, note that by construction of P_{i+1}, B_{i+1}, and N_{i+1}, the jth reduced cost of the ith projected LP is positive for all $j \in (P_i \backslash P_{i+1}) \cup (N_{i+1} \backslash N_i)$, and the jth reduced cost is zero for all $j \in P_{i+1} \backslash B_{i+1}$. Let

$$\mathbf{A}_{i+1} = \begin{bmatrix} \mathbf{c}_{i,P_{i+1}}^{\mathrm{T}} \\ \mathbf{A}_{i,P_{i+1}} \end{bmatrix} \quad \text{and} \quad \mathbf{d}_{i+1} : \mathbf{d} \mapsto \begin{bmatrix} q_i^P(\mathbf{d}) \\ \mathbf{d}_i(\mathbf{d}) \end{bmatrix}.$$

By the construction of the index set P_{i+1}, we have that

$$\frac{\mathbf{c}_{i,P_{i+1}}^{\mathrm{T}} - \mathbf{c}_{i,B_i}^{\mathrm{T}} \mathbf{A}_{i,B_i}^{-1} \mathbf{A}_{i,P_{i+1}}}{c_{i,j} - \mathbf{c}_{i,B_i}^{\mathrm{T}} \mathbf{A}_{i,B_i}^{-1} \mathbf{a}_{i,j}}$$

is the jth unit vector in $\mathbb{R}^{n_{i+1}}$ (denoted $\mathbf{e}_j^{\mathrm{T}}$), and so by Lemma 2 in Appendix A, the resulting tableau for the $(i+1)$th projected LP is

$$\begin{bmatrix} 0 & \mathbf{e}_j^{\mathrm{T}} \\ \mathbf{A}_{i,B_i}^{-1} \mathbf{d}_i(\mathbf{d}) & \mathbf{A}_{i,B_i}^{-1} \left(\mathbf{A}_{i,P_{i+1}} - \mathbf{a}_{i,j} \mathbf{e}_j^{\mathrm{T}} \right) \end{bmatrix} = \begin{bmatrix} 0 & \mathbf{e}_j^{\mathrm{T}} \\ \mathbf{A}_{i,B_i}^{-1} \mathbf{d}_i(\mathbf{d}) & \mathbf{A}_{i,B_i}^{-1} \left(\mathbf{A}_{i,P_{i+1}}^{\{j\}} \right) \end{bmatrix}. \tag{15}$$

What is important to note is that the last m_i rows of (15) form the first $n_{i+1} + 1$ columns of the tableau in assumption 4, except with the jth column equal to $\mathbf{0}$. Thus, tableau (15) is equal to

$$\begin{bmatrix} \mathbf{0} & \mathbf{E}_{i+1} \\ \mathbf{A}_{B_1}^{-1} \mathbf{d} & \mathbf{A}_{B_1}^{-1} \left(\mathbf{A}_{P_{i+1}}^{N_{i+1}} \right) \end{bmatrix}. \tag{16}$$

Similarly to the previous case, in this case the $(i+1)$th-level and projected LPs are equivalent. To see this, note that the feasible set of the $(i+1)$th projected LP (14) is equivalent to the solution set of the ith projected LP by Lemma 3 in Appendix A. By the induction assumption of equivalence and Lemma 1 in Appendix A, the equivalence of the $(i+1)$th-level and projected LPs follows.

We now optimize the $(i+1)$th projected LP (however it was constructed). The reason behind considering the projected LPs is that we can assert that after a primal simplex pivot, the new basis is $B'_{i+1} = N_{i+1} \cup B'_1$, where B'_1 is an optimal basis for the first-level LP. We also assert that $B'_k = N_k \cup B'_1$ is optimal for the kth projected LP for all $k \leq i$. Further, the tableau retains the same form, and the reduced costs of the first-level and the kth projected LPs do not change:

$$\mathbf{c}_{k,P_k}^{\mathrm{T}} - \mathbf{c}_{k,B_k}^{\mathrm{T}} \mathbf{A}_{k,B_k}^{-1} \mathbf{A}_k = \mathbf{c}_{k,P_k}^{\mathrm{T}} - \mathbf{c}_{k,B'_k}^{\mathrm{T}} \mathbf{A}_{k,B'_k}^{-1} \mathbf{A}_k \tag{17}$$

for all $k \leq i$. Since $\mathbf{d} \in F$ and the $(i+1)$th-level and projected LPs are equivalent, the primal simplex algorithm must terminate. At this point we will have optimal bases $B_k^* = N_k \cup B_1^*$, for the kth projected LP, for all $k \leq i+1$, where B_1^* is optimal for the first-level LP.

To see this, consider the specifics of a primal simplex pivot. Under any pivoting rule, let the index of the pivot column chosen be $p_c \in P_{i+1} \backslash B_{i+1}$. Note that the first $m_{i+1} - m_1$ elements of the p_cth column of the tableau (16) are zero. So to determine the pivot row we only need to consider the p_cth column of $\mathbf{A}_{B_1}^{-1}(\mathbf{A}_{P_{i+1}}^{N_{i+1}})$, but this in fact equals $\mathbf{A}_{B_1}^{-1}\mathbf{a}_{p_c}$. This means that whatever basis element is chosen to exit the basis B_{i+1} is the same element that would exit the basis B_1 if we applied the primal simplex algorithm to the first-level LP and had chosen the p_cth column as the pivot column. By assumption 3, the p_cth reduced cost of the first-level LP (given by B_1) is zero, and so this leads us to the conclusion that by following the pivot rules of the primal simplex algorithm applied to the $(i+1)$th projected LP, we are in fact executing acceptable pivots of the primal simplex algorithm applied to the first-level LP. Further, the discussion in Appendix A establishes that the reduced costs of the first-level LP will remain the same after the pivot (i.e. Eq. (17) holds for $k = 1$). Consequently, we obtain the new primal feasible basis $B'_{i+1} = N_{i+1} \cup B'_1$ for the $(i+1)$th projected LP, where B'_1 is still optimal for the first-level LP.

Similar reasoning establishes that these pivots are also acceptable primal simplex pivots applied to the kth projected LP, for *all* k. By the induction assumption 3, the p_cth reduced cost of the kth projected LP is zero, and so again all the reduced costs retain the same value after the pivot and Eq. (17) holds for $k \in \{2, \dots, i-1\}$. By construction of P_{i+1}, the p_cth reduced cost of the ith projected LP is zero, and so a similar conclusion holds for $k = i$. Again, this means $B'_k = N_k \cup B'_1$ is optimal for the kth projected LP.

Further, the tableau for the $(i+1)$th projected LP after this pivot operation has the same form as tableau (16) (just with B'_1 replacing B_1). This is because the pivot operation is executed by multiplying the tableau (from the left) by a $m_{i+1} \times m_{i+1}$ matrix of the form

$$\begin{bmatrix} \mathbf{I}_i & \mathbf{0}_i^{\mathrm{T}} \\ \mathbf{0}_i & \mathbf{Q}_1 \end{bmatrix},$$

where $\mathbf{Q}_1$ is an invertible $m_1 \times m_1$ matrix, $\mathbf{I}_i$ is the $(m_{i+1} - m_1) \times (m_{i+1} - m_1)$ identity matrix, and $\mathbf{0}_i$ is a $m_1 \times (m_{i+1} - m_1)$ matrix of zeros. If the index of the pivot row is p_r, then $\mathbf{Q}_1\mathbf{A}_{B_1}^{-1}\mathbf{a}_{p_c}$ equals the $(p_r - (m_{i+1} - m_1))$th unit vector in $\mathbb{R}^{m_1}$. This achieves the overall effect of the pivot operation, which is to change the p_cth column (of tableau (16)) into the p_rth unit vector in $\mathbb{R}^{m_{i+1}}$.

Therefore, when the simplex method terminates for the $(i+1)$th projected LP, we will have an optimal basis $B^*_k = N_k \cup B^*_1$ for the kth projected LP, for all $k \in \{1, \dots, i+1\}$, where B^*_1 is optimal for the first-level LP. All the induction assumptions hold for the $(i+1)$th step; equivalence and the form of the tableau have already been established, and induction assumption 3 holds because of how the $(i+1)$th projected LP was constructed and the reduced costs of the kth projected LPs are the same with the new bases B^*_k.

Proceeding by induction, it follows that we can obtain an optimal basis for the n_qth projected LP, $B^*_{n_q} = N_{n_q} \cup B^*_1$, where B^*_1 is an optimal basis for the first-level LP (1). The basis $B^*_{n_q}$ describes the point $\mathbf{v}^*$; by equivalence and the nature of the lexicographic LP, this point is in the solution set of the ith-level LP (2) for all i. Again by assumption 4, the only nonzero components of $\mathbf{v}^*$ are those corresponding to B_1^*, so we have $\mathbf{c}_i^{\mathrm{T}}\mathbf{v}^* = \mathbf{c}_{i,B_1^*}^{\mathrm{T}}\mathbf{A}_{B_1^*}^{-1}\mathbf{d}$ for all i. So we have that Eq. (12) holds.

We now establish the final claim that Eq. (12) holds for all $\mathbf{d}$ such that B_1^* is optimal. The reasoning follows from the previous argument, although formally a separate induction argument is needed. The essence of the argument is that the basis $B_i^* = N_i \cup B_1^*$ is optimal for the corresponding projected LP as defined earlier for all $\mathbf{d}$ such that B_1^* is optimal for the first-level LP. This is because dual feasibility for each basis does not change, while the form of the tableau from induction assumption 4 indicates that primal feasibility of B_1^* implies primal feasibility of B_i^*. Further, if the ith-level and projected LPs are equivalent for all $\mathbf{d}$ such that B_1^* is optimal, then the $(i+1)$th-level and projected LPs are equivalent for all $\mathbf{d}$ such that B_1^* is optimal. This follows from application of Lemma 1 and, if necessary, Lemma 3 in Appendix A, which indicates that null variables remain null variables for all $\mathbf{d}$ such that B_i^* is optimal. Combined with the previous observation, this means that the $(i+1)$th-level and projected LPs are equivalent for all $\mathbf{d}$ such that B_1^* is optimal. If the construction terminated early after determining that the ith projected LP has a unique solution, then this projected LP has a unique solution for as long as the basis B_i^* is optimal, which again holds for all $\mathbf{d}$ such that B_1^* is optimal. The conclusion of the induction argument is that for all $\mathbf{d}$ such that B_1^* is optimal, it describes a point in the solution set of each projected LP, and by equivalence, a point in the solution set of each level of the lexicographic LP (1)–(2). □

6 Examples

The simple example from Sect. 3 is reconsidered to clarify the qualitative difference between Algorithm 1 and the previously mentioned direct and time-stepping methods. Then, two examples based on dynamic flux balance analysis are presented. In Sect. 6.2, a model of batch fermentation displaying domain issues is presented. This example also demonstrates a significant numerical difference between the performance of Algorithm 1 and the direct method. In Sect. 6.3, a model of batch fermentation is presented in which a non-unique solution set of the embedded LP is encountered. The LP is reformulated as a lexicographic LP to resolve the non-uniqueness to obtain a better-defined and more numerically tractable problem.

6.1 Robustness for simple example

Consider once more the simple example from Sect. 3.1. The solution estimate after an explicit Euler step (of stepsize h) is still $\widetilde{\mathbf{x}}(h) = (h, 0)$. As in Sect. 3.1, $\widetilde{\mathbf{x}}(h) \notin K = \{\mathbf{z} : z_2 \geq z_1^2\}$. However, in contrast with the direct method, this is not a complication; at any time t, the system of equations to be solved for the DAE reformulation from Algorithm 1 is

$$\begin{aligned}\widetilde{\mathbf{x}}(t+h) - \widetilde{\mathbf{x}}(t) - h\mathbf{f}(\widetilde{\mathbf{x}}(t), q^B(\widetilde{\mathbf{u}}_B(t))) &= \mathbf{0},\\ \mathbf{A}_B\widetilde{\mathbf{u}}_B(t) - \mathbf{b}(\widetilde{\mathbf{x}}(t)) &= \mathbf{0},\end{aligned}$$

where q^B is defined as in Algorithm 1. Whatever the choice of the basis B is, $\mathbf{u}_B(t)$ and q^B are well defined and the system of equations has a solution. This is a significant qualitative difference between Algorithm 1 and the direct or time-stepping methods.

Of course, this qualitative difference translates to a noticeable difference in numerical performance. When the solution is at the (numerical) boundary of K, only Algorithm 1 can guarantee that an approximate solution can be continued. As demonstrated by the next example, this can lead to an unmistakable difference in the quality of the numerical solution. Specifically, the direct method fails or gives an incorrect indication of when the solution of the extended ODE is no longer a solution of the original system (4).

6.2 *E. coli* fermentation

Batch and fed-batch fermentation reactions are important industrial processes for the production of valuable chemicals such as ethanol. This example considers a model of a fermentation reactor consisting of the dynamic mass balances of the reactor coupled to a genome-scale network reconstruction of the *E. coli* metabolism presented in [14]. Using information gleaned from genomic analysis, *E. coli*'s metabolism can be modeled as a network of reactions that must satisfy simple stoichiometric constraints. Analysis and construction of such a network is called flux balance analysis (FBA) [28]. However, this network is often under-determined; the fluxes of the different substrates and metabolites can vary and still produce a system that satisfies the stoichiometric constraints. Thus, one assumes that fluxes will be such that some cellular objective is maximized. Most often, the production of biomass is chosen as the cellular objective to maximize, and in general it is a reasonable choice [29]. The result, then, is in fact a system that has the same form as (4). The simulation represents the initial phase of batch operation of the fermentation reactor under aerobic growth on glucose and xylose media. No ethanol production during aerobic conditions is observed; this phase is used to increase the biomass. Thus, the concentration of ethanol is omitted from the dynamics.

6.2.1 Model

The dynamic mass balance equations of the extracellular environment of the batch reactor are

$$\begin{aligned} \dot{x}(t) &= \mu(t)x(t), \\ \dot{g}(t) &= -u_g(t)x(t), \\ \dot{z}(t) &= -u_z(t)x(t), \end{aligned} \tag{18}$$

where $\mathbf{x}(t) = (x(t), g(t), z(t))$ is the vector of biomass, glucose and xylose concentrations, respectively, at time t. The uptake kinetics for glucose, xylose and oxygen are given by the Michaelis–Menten kinetics

$$u_g(t) = u_{g,max}\frac{g(t)}{K_g + g(t)}, \tag{19}$$

$$u_z(t) = u_{z,max}\frac{z(t)}{K_z + z(t)}\frac{1}{1 + \frac{g(t)}{K_{ig}}}, \tag{20}$$

$$u_o(t) = u_{o,max}\frac{o(t)}{K_o + o(t)}, \tag{21}$$

where $u_{g,max}$, $u_{z,max}$, $u_{o,max}$, K_g, K_z, K_o, and K_{ig} are known constant parameters. It is assumed that the oxygen concentration in the reactor, $o(t)$, is controlled and therefore a known value. Meanwhile, the growth rate $\mu(t)$ is determined from the metabolic network model of the *E. coli* bacterium iJR904 [34], which is available online [37]. The model consists of 625 unique metabolites, 931 intracellular fluxes, 144 exchange fluxes and an additional flux representing the biomass generation as growth rate $\mu(t)$. The flux balance model is an LP of the form

$$\begin{aligned} \mu(t) = \min_{\mathbf{v}\in\mathbb{R}^{n_v}} \ & \mathbf{c}^{\mathrm{T}}\mathbf{v} \\ \text{s.t. } & \mathbf{S}\mathbf{v} = \mathbf{0}, \\ & v_{g_{ext}} = u_g(t), \\ & v_{z_{ext}} = u_z(t), \\ & v_{o_{ext}} = u_o(t), \\ & \mathbf{v}^{LB} \le \mathbf{v} \le \mathbf{v}^{UB}, \end{aligned} \tag{22}$$

where n_v is the number of fluxes, n_m is the number of metabolites, $\mathbf{S} \in \mathbb{R}^{n_m \times n_v}$ is the stoichiometry matrix of the metabolic network, $\mu(t)$ is the growth rate and $\mathbf{v}^{LB}$ and $\mathbf{v}^{UB}$ are the lower and upper bounds on the fluxes. The metabolic network is connected to the extracellular environment through the exchange fluxes for glucose, xylose and oxygen $v_{g_{ext}}$, $v_{z_{ext}}$ and $v_{o_{ext}}$, respectively, which are given by Eqs. (19)–(21). After putting the LP (22) in standard form and assuring that it satisfies Assumption 1, the LP has 749 constraints and 2150 primal variables.

6.2.2 Simulation results

The solution of the system (18)–(22) was calculated by the DSL48LPR implementation of Algorithm 1 and, for comparison, by the direct method, which was implemented with DSL48E (without any events) with the function evaluator calling CPLEX.

All numerical parameter values including the initial conditions are according to [14]. The time evolution of the dynamic states is shown in Fig. 1. First glucose, as the preferred carbon source, is consumed. After glucose has been depleted, at around 7h, the optimal basis changes and xylose becomes the main carbon source. The final batch time is determined by the glucose and xylose concentrations. The simulation stops when glucose and xylose concentration are equal to zero (around 8.2h); at this point, the LP is infeasible and so by Corollary 1 the solution ceases to exist. This makes sense physically, since with no carbon source the *E. coli* stop growing and begin to die; cell death is not a phase that the flux balance model can really predict and so the simulation must stop.

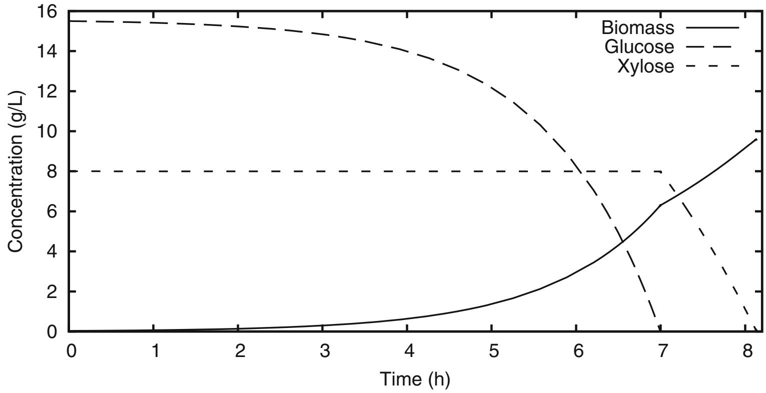

Fig. 1 Species concentrations from Eqs. (18)–(22) in bioreactor as calculated by DSL48LPR

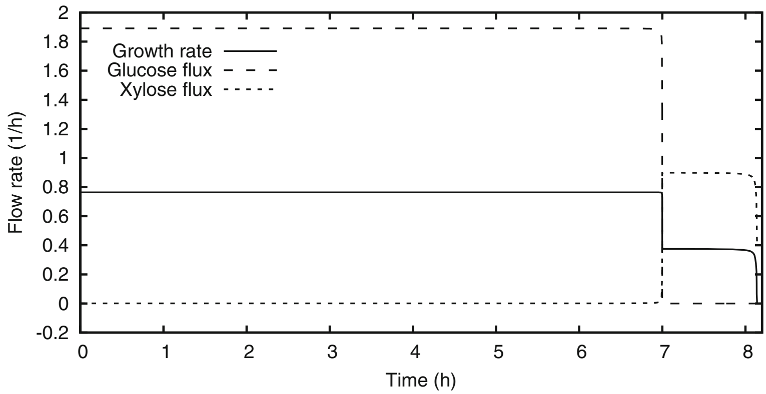

Fig. 2 A representative selection of exchange fluxes (solution of LP (22)) as calculated by DSL48LPR. Note the extremely steep, but still continuous, change at around 7h, when the metabolism changes

When simulating the system with DSL48E and CPLEX, the simulation fails at the point when the *E. coli* switches from glucose to xylose metabolism. This is clear when examining the primal variables (the fluxes) in Fig. 2. The values of the primal variables change quite rapidly (however they are still continuous). This indicates that the system (18)–(22) is stiff. Stiff dynamics combined with the numerical manifestation of domain issues as discussed in Sect. 3 cause the direct method to fail. In contrast, DSL48LPR manages to integrate past the change in metabolism and more accurately indicate when the solution fails to exist.

6.2.3 Computational times

This example also provides a good chance to compare the computational times for various solution methods. The time required by DSL48LPR and by various forms of the direct method to complete the simulation are compared in Table 1. The direct method was implemented using various different LP algorithms, and this can impact the solution time quite strongly. DSL48LPR is fast, both on the interval [0, 7]h and on the whole simulation interval. Meanwhile, DSL48E embedding CPLEX fails to complete the entire simulation, but the computational time to run the simulation to the point of failure can vary quite a lot. Using dual simplex with an advanced basis is the fastest, and competitive with DSL48LPR. This follows from the fact that using a dual feasible basis to warm start dual simplex is very similar to the basic algorithm of DSL48LPR. While this basis is also optimal, CPLEX only needs to solve a linear system to determine the values of the primal variables given the new value of the right-hand side vector. It should be noted that, to the authors' knowledge, this use of dual simplex has not been proposed before for the solution of ODEs with LPs embedded.

The other LP algorithms, however, increase the simulation time. Neglecting that a dual feasible basis is available and using full (Phase I and Phase II) simplex is slower, followed by a barrier method (most likely the primal-dual path following algorithm, see §9.5 of [5]). Although interior point methods for LPs are praised for their polynomial solution time, it is an unwise choice in this context. Comparable to a nonlinear solve in at least 2000 variables, it incurs much more overhead, likely because it is factoring the necessary matrices more often than DSL48E is factoring the Jacobian within DSL48LPR. Further, it is possible that there are issues initializing the algorithm, since the previous solution point may be infeasible after a perturbation of the value of **b**; consequently, the algorithm again lacks advanced starting point information which slows it down considerably.

6.3 Yeast fermentation

Normally, the solution sets of flux-balance models are not singletons [25]. Consider a second dynamic flux balance simulation of fed-batch fermentation using *Saccharomyces cerevisiae*. Besides ethanol, as the main metabolic product of interest, other by-products, such as glycerol, can be analyzed. A non-unique glycerol flux is predicted by the metabolic network reconstruction iND750 [8] of *S. cerevisiae* under anaerobic growth conditions [16]. In order to determine the range of the glycerol flux during batch fermentation, this example utilizes a lexicographic LP to determine a maximum and then minimum glycerol flux at the optimal growth rate.

This model has been considered in [17] for the production of ethanol by fed-batch fermentation of *S. cerevisiae*. The dynamics are

$$
\begin{aligned}
\dot{v}(t) &= d(t), \\
\dot{g}(t) &= -u_g(t)x(t) + d(t)(g_{in} - g(t))/v(t), \\
\dot{x}(t) &= u_b(t)x(t) - d(t)x(t)/v(t), \\
\dot{e}(t) &= u_e(t)x(t) - d(t)e(t)/v(t), \\
\dot{h}(t) &= u_h(t)x(t) - d(t)h(t)/v(t),
\end{aligned}
\tag{23}
$$

where $v(t)$ is the total volume in the reactor, $d(t)$ is the dilution rate, and $g(t)$, $x(t)$, $e(t)$ and $h(t)$ are the concentrations of glucose, biomass, ethanol and glycerol respectively, in the reactor. Meanwhile, g_{in} is the constant glucose inlet concentration, u_g is again given by (19), and $u_b(t)$, $u_e(t)$, $u_h(t)$ are given by

Table 1 Computational times (averaged over 50 runs performed on a 32-bit Linux virtual machine allocated a single core of a 3.07 GHz Intel Xeon CPU) and integration statistics for solving Eqs. (18)–(22) with various methods

Method	DSL48LPR	DSL48E embedding CPLEX		
		Dual simplex	Full simplex	Barrier method
CPU time, full simulation (s)	1.196	[a]	[a]	[a]
CPU time, on [0, 7]h (s)	1.004	0.436	2.799	4.772
Integration steps on [0, 7]h	408	125	125	125
Jacobian evaluations on [0, 7]h	169	53	53	53
Error test failures on [0, 7]h	30	22	22	22
Convergence test failures on [0, 7]h	0	11	11	11

[a] Indicates that the method failed before finishing the simulation

$$\begin{aligned} u_b(t) = \max_{\mathbf{v}}\ & v_b \\ \text{s.t. }\ & \mathbf{A}\mathbf{v} = \mathbf{b}(g(t), e(t), o(t)), \\ & \mathbf{v} \geq \mathbf{0}, \end{aligned} \tag{24}$$

$$\begin{aligned} u_e(t) = \max_{\mathbf{v}}\ & v_e \\ \text{s.t. }\ & \mathbf{A}\mathbf{v} = \mathbf{b}(g(t), e(t), o(t)), \\ & v_b = u_b(t), \\ & \mathbf{v} \geq \mathbf{0}, \end{aligned} \tag{25}$$

and

$$\begin{aligned} u_h(t) = \max_{\mathbf{v}}\ & v_h \\ \text{s.t. }\ & \mathbf{A}\mathbf{v} = \mathbf{b}(g(t), e(t), o(t)), \\ & v_e = u_e(t), \\ & v_b = u_b(t), \\ & \mathbf{v} \geq \mathbf{0}. \end{aligned} \tag{26}$$

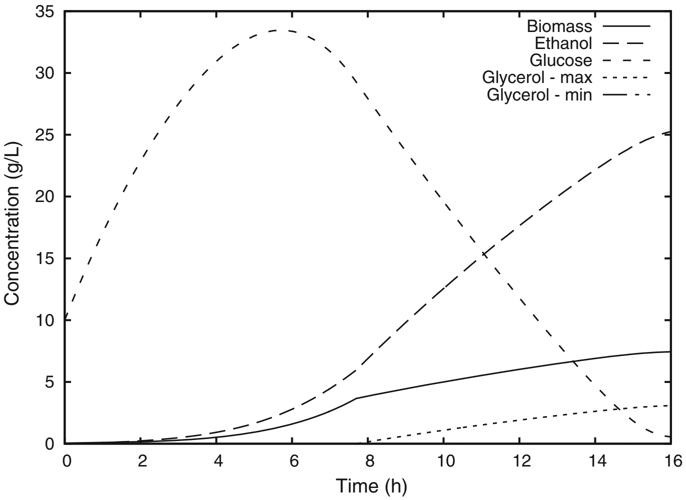

Fig. 3 Species concentrations from Eq. (23) in bioreactor as calculated by DSL48LPR. Note that the glycerol concentration potentially can take a range of values, if the glycerol flux is not explicitly fixed to a maximal or minimal value (minimal value is zero)

The LP (24) is obtained by transforming a flux balance model for yeast in a similar manner to what was done in the previous example; (24) is connected to the extracellular environment via the Michaelis–Menten equations (19) and (21), and then put into standard form. Note that $(u_b(t), u_e(t), u_h(t))$ is the solution to a lexicographic LP. After maximizing the growth rate, the optimal growth rate is added as a constraint and the resulting program is optimized with respect to ethanol flux. This optimal ethanol flux is again added as a constraint and then glycerol flux is maximized. The result is that these three fluxes are now uniquely defined and the problem (23) is well-defined. It is more difficult to address the non-uniqueness of the glycerol flux when solving (23) with the direct method; even if it is considered it requires the solution of extra LPs which can be costly. Meanwhile, a lexicographic LP provides a more straightforward way to enforce uniqueness, which reduces the ambiguity of the simulation results.

The parameter values for the simulation can be found in [17]. The simulation presents an aerobic-anaerobic operation. The aerobic to anaerobic switch occurs at 7.7h, after which a range of glycerol flux rates are possible. This leads to a maximum and minimum possible glycerol concentration; the discrepancy is called the production envelope [25]. To determine this envelope, a second simulation in which glycerol flux is instead minimized in (26) is performed. This simulation shows no glycerol production throughout the batch reaction. At the end of the simulations, the difference between the maximum and minimum glycerol concentrations is 3.71 g/L, where the concentrations of nutrients and metabolites are on the order of 10 g/L throughout the simulation. Clearly, a non-unique solution of the LP can have a significant impact on the overall solution of the dynamic system. The results are seen in Fig. 3.

7 Conclusions

This work has analyzed the initial value problem in ordinary differential equations with a parametric lexicographic linear program embedded. This problem finds application in dynamic flux balance analysis, which is used in the modeling of industrial fermentation reactions. This work has proposed a numerical method which has distinct advantages over other applicable methods. These advantages allow the method to be successfully applied to examples of DFBA, and achieve unambiguously improved approximate solutions to these examples. The current implementation of the proposed method proves very successful in the motivating application of DFBA. Furthermore, the method is flexible and allows various numerical integration routines to be applied.

Acknowledgments The authors thank Novartis Pharmaceuticals as part of the Novartis-MIT Center for Continuous Manufacturing for funding this research.

Appendix A: Supporting results for lexicographic optimization

This section presents some definitions, results, and discussion to support the proof of Theorem 2, which deals with finding a specific optimal basis for the lexicographic LP.

Definition 1 Equivalence.

1. Two sets $S_1 \subset \mathbb{R}^{n_1}$ and $S_2 \subset \mathbb{R}^{n_2}$ with $n_1 \leq n_2$ are equivalent if $n_1 < n_2$ and $S_2 = S_1 \times \{\mathbf{0}\}$, or $n_1 = n_2$ and $S_2 = S_1$.
2. Two linear programs are equivalent if their solution sets are equivalent.

Intuitive results regarding equivalence follow.

Lemma 1 1. *Assume $n_1 \leq n_2 \leq n_3$. Let $F_i \subset \mathbb{R}^{n_i}$ for $i \in \{1, 2, 3\}$. If sets F_1 and F_2 are equivalent and F_2 and F_3 are equivalent, then F_1 and F_3 are equivalent.*
2. *If two sets $F_1 \in \mathbb{R}^{n_1}$ and $F_2 \in \mathbb{R}^{n_2}$ with $n_1 \leq n_2$ are equivalent, then the linear programs*

$$\min\{\mathbf{c}^{\mathrm{T}}\mathbf{v} : \mathbf{v} \in F_1\} \quad \textit{and} \quad \min\{\widehat{\mathbf{c}}^{\mathrm{T}}\mathbf{v} : \mathbf{v} \in F_2\}$$

are equivalent, where $\widehat{\mathbf{c}} = (\mathbf{c}, \widetilde{\mathbf{c}})$ for any $\mathbf{c} \in \mathbb{R}^{n_1}$ and $\widetilde{\mathbf{c}} \in \mathbb{R}^{n_2 - n_1}$.

For the next two results refer to the lexicographic LP

$$q(\mathbf{d}) = \inf\{\mathbf{c}^{\mathrm{T}}\mathbf{v} : \mathbf{M}\mathbf{v} = \mathbf{d}, \mathbf{v} \geq \mathbf{0}\}, \tag{27}$$

$$\widehat{q}(\mathbf{d}) = \inf\{\widehat{\mathbf{c}}^{\mathrm{T}}\mathbf{v} : \mathbf{M}\mathbf{v} = \mathbf{d}, \mathbf{c}^{\mathrm{T}}\mathbf{v} = q(\mathbf{d}), \mathbf{v} \geq \mathbf{0}\}. \tag{28}$$

The next result establishes the form of the simplex tableau for the two-level lexicographic LP. Strictly speaking, tableau (29) below is missing the "zeroth" row of reduced costs for the second-level LP (28); for simplicity it is omitted.

Lemma 2 *Consider the lexicographic LP (27)–(28). Let B be a dual feasible basis for the first-level LP (27), and assume that the jth reduced cost is positive ($c_j -$*

$\mathbf{c}_B^{\mathrm{T}}\mathbf{M}_B^{-1}\mathbf{m}_j > 0$). *For all* $\mathbf{d}$ *such that* B *is optimal for the first-level LP, the simplex tableau for the second-level LP* (28) *resulting from the basis* $\widehat{B} = \{j\} \cup B$ *is*

$$\begin{bmatrix} c_j & \mathbf{c}_B^{\mathrm{T}} \\ \mathbf{m}_j & \mathbf{M}_B \end{bmatrix}^{-1} \begin{bmatrix} q(\mathbf{d}) & \mathbf{c}^{\mathrm{T}} \\ \mathbf{d} & \mathbf{M} \end{bmatrix} = \begin{bmatrix} 0 & \frac{\mathbf{c}^{\mathrm{T}} - \mathbf{c}_B^{\mathrm{T}}\mathbf{M}_B^{-1}\mathbf{M}}{c_j - \mathbf{c}_B^{\mathrm{T}}\mathbf{M}_B^{-1}\mathbf{m}_j} \\ \mathbf{M}_B^{-1}\mathbf{d} & \mathbf{M}_B^{-1}\left(\mathbf{M} - \mathbf{m}_j \frac{\mathbf{c}^{\mathrm{T}} - \mathbf{c}_B^{\mathrm{T}}\mathbf{M}_B^{-1}\mathbf{M}}{c_j - \mathbf{c}_B^{\mathrm{T}}\mathbf{M}_B^{-1}\mathbf{m}_j}\right) \end{bmatrix}. \quad (29)$$

Proof The proof proceeds by using Schur complements to form the inverse of $\begin{bmatrix} c_j & \mathbf{c}_B^{\mathrm{T}} \\ \mathbf{m}_j & \mathbf{M}_B \end{bmatrix}$, performing the matrix multiplication, and simplifying, noting that $\mathbf{c}_B^{\mathrm{T}}\mathbf{M}_B^{-1}\mathbf{d} = q(\mathbf{d})$ for all $\mathbf{d}$ such that $\mathbf{M}_B^{-1}\mathbf{d} \geq \mathbf{0}$. □

The concept of a "null variable" is important to the proof of Theorem 2, which is defined as a variable which is zero everywhere in the feasible set of an LP. It is clear that removing a null variable and the corresponding parameters (components of the cost vector and columns of the constraint matrix) yields an equivalent LP. The next result states a way to identify null variables in the second-level LP (28).

Lemma 3 *Consider the lexicographic LP* (27)–(28). *Let* B *be a dual feasible basis for the first-level LP* (27), *and assume that the* j*th reduced cost is positive* ($c_j - \mathbf{c}_B^{\mathrm{T}}\mathbf{M}_B^{-1}\mathbf{m}_j > 0$). *For all* $\mathbf{d}$ *such that* B *is optimal for the first-level LP*, $v_j = 0$ *for all* $\mathbf{v}$ *feasible in the second-level LP* (28).

Proof The result follows from the "null variable theorem" in §4.7 of [24]; this states that v_j is a null variable for a general standard-form LP (27) if and only if there exists a nonzero $\mathbf{p}$ such that $\mathbf{p}^{\mathrm{T}}\mathbf{d} = 0$, $\mathbf{p}^{\mathrm{T}}\mathbf{M} \geq \mathbf{0}^{\mathrm{T}}$, and the jth component of $\mathbf{p}^{\mathrm{T}}\mathbf{M}$ is strictly greater than zero. Applying this result to the second-level LP (28), the result follows from inspection of the tableau (29); the first row of $\begin{bmatrix} c_j & \mathbf{c}_B^{\mathrm{T}} \\ \mathbf{m}_j & \mathbf{M}_B \end{bmatrix}^{-1}$ serves as the appropriate $\mathbf{p}$. □

Finally, some aspects of the primal simplex algorithm are noted. If one has a optimal basis already, but a different optimal basis is sought, a pivot could be forced in the sense that, while the ith reduced cost is zero, the ith column is chosen as the pivot column. If the ith reduced cost is zero, but the pivot operation is carried out in the standard way, a new primal feasible basis is obtained for which the reduced costs are the same as the old basis, and so the new basis is also optimal. This is because the reduced costs are updated in a pivot operation by adding a multiple of the pivot row to the reduced costs (the zeroth row of the tableau) so that the ith entry of the zeroth row is zero. But, if the ith reduced cost is already zero, no changes to the zeroth row are made, and so the reduced costs retain the same values.

Appendix B: Domain issues and time-stepping

This section presents an example demonstrating that time-stepping methods such as those mentioned in Sect. 3.2 still suffer from domain issues. More generally, this example shows that implicit integration methods also suffer from domain issues.

Consider an example similar to the one in Sect. 3.1:

$$\mathbf{x}(0) = \mathbf{0}, \quad \dot{\mathbf{x}}(t) = \mathbf{f}\left(\mathbf{x}(t), q(\mathbf{x}(t))\right) = \begin{bmatrix} 1 \\ x_2(t)q(\mathbf{x}(t)) - (x_2(t))^2 + 2x_1(t) \end{bmatrix}, \tag{30}$$

where $q(\mathbf{z}) = \min\{v : z_2 \le v \le z_1^2\}$.

The embedded LP is feasible only if $\mathbf{x} \in K = \{\mathbf{z} : z_2 \le z_1^2\}$, a closed, nonconvex set. Note that $\mathbf{x}(t) = (t, t^2)$ is a solution. Letting $\mathbf{b}(\mathbf{z}) = (z_1^2, -z_2)$ and rewriting q in terms of the embedded LP's dual, one has

$$q(\mathbf{z}) = \max\{\mathbf{b}(\mathbf{z})^{\mathrm{T}}\mathbf{w} : \mathbf{w} \le 0, w_1 - w_2 = 1\}.$$

Letting W denote the feasible set of the above (dual) LP, this is equivalent to finding $\mathbf{w}^* \in W$ such that $(\mathbf{w} - \mathbf{w}^*)^{\mathrm{T}}(-\mathbf{b}(\mathbf{z})) \ge 0$, $\forall \mathbf{w} \in W$. This is a parametric variational inequality and is denoted $\mathrm{VI}(W, -\mathbf{b}(\mathbf{z}))$. This requires the dynamics to be rewritten as

$$\begin{aligned} &\mathbf{f}\left(\mathbf{x}(t), q(\mathbf{x}(t))\right) \\ &\quad = \widehat{\mathbf{f}}(\mathbf{x}(t), \mathbf{u}(t)) = \begin{bmatrix} 1 \\ x_2(t)\left((x_1(t))^2 u_1(t) - x_2(t)u_2(t)\right) - (x_2(t))^2 + 2x_1(t) \end{bmatrix}, \end{aligned}$$

where $\mathbf{u}(t)$ is the solution of $\mathrm{VI}(W, -\mathbf{b}(\mathbf{x}(t)))$. Given $h > 0$, an implicit time-stepping scheme takes the form

$$\begin{aligned} \widetilde{\mathbf{x}}(t+h) &= \widetilde{\mathbf{x}}(t) + h\widehat{\mathbf{f}}\left(\widetilde{\mathbf{x}}(t+h), \widetilde{\mathbf{u}}(t+h)\right), \\ \widetilde{\mathbf{u}}(t+h) &\text{ solves } \mathrm{VI}(W, -\mathbf{b}(\widetilde{\mathbf{x}}(t+h))). \end{aligned} \tag{31}$$

Typically this implicit system is solved as the equivalent variational inequality $\mathrm{VI}(\mathbb{R}^2 \times W, \mathbf{g}^t)$ where

$$\mathbf{g}^t(\mathbf{z}, \mathbf{v}) = \begin{bmatrix} z_1 - \widetilde{x}_1(t) - h \\ z_2 - \widetilde{x}_2(t) - h(z_1^2 z_2 v_1 - z_2^2 v_2 - z_2^2 + 2z_1) \\ -z_1^2 \\ z_2 \end{bmatrix}$$

(see for instance [30]). However, again letting $\widetilde{\mathbf{x}}(0) = \mathbf{x}(0) = \mathbf{0}$, the initial variational inequality $\mathrm{VI}(\mathbb{R}^2 \times W, \mathbf{g}^0)$ does *not* have a solution for any choice of h.

To see this, assume, for a contradiction, that a solution exists. Then there is a $(\mathbf{z}^*, \mathbf{v}^*) \in \mathbb{R}^2 \times W$ such that

$$\begin{aligned} &(z_1 - z_1^*)(z_1^* - h) + (z_2 - z_2^*)\left(z_2^* - h((z_1^*)^2 z_2^* v_1^* - (z_2^*)^2 v_2^* - (z_2^*)^2 + 2z_1^*)\right) \\ &\quad + (v_1 - v_1^*)(-(z_1^*)^2) + (v_2 - v_2^*)(z_2^*) \ge 0, \end{aligned} \tag{32}$$

for all $(\mathbf{z}, \mathbf{v}) \in \mathbb{R}^2 \times W$. First note that $z_1^* = h$, otherwise one could always find a $z_1 \in \mathbb{R}$ such that the inequality (32) did not hold. Similarly, one must have

$$z_2^* = h\left((z_1^*)^2 z_2^* v_1^* - (z_2^*)^2 v_2^* - (z_2^*)^2 + 2z_1^*\right). \tag{33}$$

Using this in inequality (32), one obtains

$$(v_1 - v_1^*)(-h^2) + (v_2 - v_2^*)(z_2^*) \geq 0,$$

for all $(\mathbf{z}, \mathbf{v}) \in \mathbb{R}^2 \times W$. For any $\mathbf{v} \in W$, one can write $v_2 = v_1 - 1$. Then one gets

$$(v_1 - v_1^*)(-h^2) + (v_1 - 1 - (v_1^* - 1))(z_2^*) \geq 0,$$

which yields

$$(v_1 - v_1^*)(z_2^* - h^2) \geq 0,$$

for all $v_1 \leq 0$, where $v_1^* \leq 0$ and $z_2^* \in \mathbb{R}$ satisfy

$$hv_1^*(z_2^*)^2 + (1 - h^3 v_1^*)z_2^* - 2h^2 = 0 \tag{34}$$

(which is obtained from Eq. (33) via the substitutions $z_1^* = h$ and $v_2^* = v_1^* - 1$).

One can now analyze three cases:

1. $z_2^* > h^2$: However, if this was the case, then whatever the value of v_1^*, one could always find a $v_1' < v_1^*$ which then implies $(v_1' - v_1^*)(z_2^* - h^2) < 0$, which is a contradiction.
2. $z_2^* < h^2$: However, if this was the case, one must have $v_1^* = 0$, otherwise there exists a v_1' such that $v_1^* < v_1' \leq 0$ which then implies that $(v_1' - v_1^*)(z_2^* - h^2) < 0$. Thus, assuming $v_1^* = 0$, use Eq. (34) to check the value of z_2^*. However, that yields $z_2^* = 2h^2$, which contradicts $z_2^* < h^2$.
3. $z_2^* = h^2$: However, if this was the case, one can use Eq. (34) to check the consistency of values. This yields

$$h^5 v_1^* + h^2 - h^5 v_1^* - 2h^2 = 0 \implies -h^2 = 0,$$

which contradicts $h > 0$.

Thus, it follows that there does not exist a point $(\mathbf{z}^*, \mathbf{v}^*) \in \mathbb{R}^2 \times W$ which solves $\mathrm{VI}(\mathbb{R}^2 \times W, \mathbf{g}^0)$.

Note that the implicit time-stepping scheme (31) is equivalent to the direct method applying an implicit Euler step to the original system (30). Thus, the failure of the system (31) to have a solution indicates that the direct method, even with an implicit integration routine, also fails.

References

1. Acary, V.: Time-stepping via complementarity. In: Vasca, F., Iannelli, L. (eds.) Dynamics and Control of Switched Electronic Systems. Advances in Industrial Control, chap. 14, pp. 417–450. Springer, London (2012)
2. Anitescu, M., Tasora, A.: An iterative approach for cone complementarity problems for nonsmooth dynamics. Comput. Optim. Appl. **47**(2), 207–235 (2010)
3. Aubin, J.P.: Viability Theory. Birkhäuser, Boston (1991)
4. Barton, P.I., Lee, C.K.: Modeling, simulation, sensitivity analysis, and optimization of hybrid systems. ACM Trans. Model. Comput. Simul. **12**(4), 256–289 (2002)
5. Bertsimas, D., Tsitsiklis, J.N.: Introduction to Linear Optimization. Athena Scientific, Belmont (1997)
6. Brenan, K.E., Campbell, S.L., Petzold, L.R.: Numerical Solution of Initial-Value Problems in Differential-Algebraic Equations. SIAM, Philadelphia (1996)
7. Dontchev, A.L., Lempio, F.: Difference methods for differential inclusions: a survey. SIAM Rev. **34**(2), 263–294 (1992)
8. Duarte, N.C., Herrgård, M.J., Palsson, B.Ø.: Reconstruction and validation of *Saccharomyces cerevisiae* iND750, a fully compartmentalized genome-scale metabolic model. Genome Res. **14**(7), 1298–1309 (2004)
9. Duff, I.S., Reid, J.K.: The design of MA48: a code for the direct solution of sparse unsymmetric linear systems of equations. ACM Trans. Math. Softw. **22**(2), 187–226 (1996)
10. Filippov, A.F.: Differential Equations with Discontinuous Righthand Sides. Kluwer Academic, Boston (1988)
11. Gal, T.: Postoptimal Analyses, Parametric Programming, and Related Topics, 2nd edn. Walter de Gruyter, New York (1995)
12. Galán, S., Feehery, W.F., Barton, P.I.: Parametric sensitivity functions for hybrid discrete/continuous systems. Appl. Numer. Math. **31**(1), 17–47 (1999)
13. Han, L., Tiwari, A., Camlibel, M.K., Pang, J.S.: Convergence of time-stepping schemes for passive and extended linear complementarity systems. SIAM J. Numer. Anal. **47**(5), 3768–3796 (2009)
14. Hanly, T.J., Henson, M.A.: Dynamic flux balance modeling of microbial co-cultures for efficient batch fermentation of glucose and xylose mixtures. Biotechnol. Bioeng. **108**(2), 376–85 (2011)
15. Hanly, T.J., Henson, M.A.: Dynamic metabolic modeling of a microaerobic yeast co-culture: predicting and optimizing ethanol production from glucose/xylose mixtures. Biotechnol. Biofuels **6**(1), 44 (2013)
16. Hjersted, J.L., Henson, M.A.: Steady-state and dynamic flux balance analysis of ethanol production by *Saccharomyces cerevisiae*. IET Syst. Biol. **3**(3), 167–179 (2009)
17. Hjersted, J.L., Henson, M.A., Mahadevan, R.: Genome-scale analysis of *Saccharomyces cerevisiae* metabolism and ethanol production in fed-batch culture. Biotechnol. Bioeng. **97**(5), 1190–1204 (2007)
18. Hoeffner, K., Harwood, S.M., Barton, P.I.: A reliable simulator for dynamic flux balance analysis. Biotechnol. Bioeng. **110**(3), 792–802 (2013)
19. Ignizio, J.P.: Goal Programming and Extensions. Lexington Books, Lexington (1976)
20. Isermann, H.: Linear lexicographic optimization. OR Spektrum **4**(4), 223–228 (1982)
21. Kaplan, U., Türkay, M., Biegler, L.T., Karasözen, B.: Modeling and simulation of metabolic networks for estimation of biomass accumulation parameters. Discrete Appl. Math. **157**(10), 2483–2493 (2009)
22. Korhonen, P., Halme, M.: Using lexicographic parametric programming for searching a non-dominated set in multiple-objective linear programming. J. Multi-Criteria Decis. Anal. **5**, 291–300 (1996)
23. Lambert, J.D.: Numerical Methods for Ordinary Differential Systems: The Initial Value Problem. Wiley, New York (1991)
24. Luenberger, D.G.: Linear and Nonlinear Programming, 2nd edn. Addison-Wesley, Reading (1984)
25. Mahadevan, R., Schilling, C.H.: The effects of alternate optimal solutions in constraint-based genome-scale metabolic models. Metab. Eng. **5**(4), 264–276 (2003)
26. Mattheij, R., Molenaar, J.: Ordinary Differential Equations in Theory and Practice. SIAM, Philadelphia (2002)
27. Mitsos, A., Barton, P.I.: Parametric mixed-integer 0-1 linear programming: the general case for a single parameter. Eur. J. Oper. Res. **194**, 663–686 (2009)
28. Orth, J.D., Thiele, I., Palsson, B.Ø.: What is flux balance analysis? Nat. Biotechnol. **28**(3), 245–248 (2010)
29. Palsson, B.: Systems Biology: Properties of Reconstructed Networks. Cambridge University Press, New York (2006)

30. Pang, J.S., Stewart, D.E.: Differential variational inequalities. Math. Program. Ser. A **113**, 345–424 (2008)
31. Park, T., Barton, P.I.: State event location in differential-algebraic models. ACM Trans. Model. Comput. Simul. **6**(2), 137–165 (1996)
32. Pourkarimi, L., Zarepisheh, M.: A dual-based algorithm for solving lexicographic multiple objective programs. Eur. J. Oper. Res. **176**(3), 1348–1356 (2007)
33. Raghunathan, A.U., Perez-Correa, J.R., Agosin, E., Biegler, L.T.: Parameter estimation in metabolic flux balance models for batch fermentation—formulation & solution using differential variational inequalities. Ann. Oper. Res. **148**(1), 251–270 (2006)
34. Reed, J., Vo, T., Schilling, C., Palsson, B.Ø.: An expanded genome-scale model of *Escherichia coli* K-12 (iJR904 GSM/GPR). Genome Biol. **4**(9), R54 (2003)
35. Robinson, S.M., Day, R.H.: A sufficient condition for continuity of optimal sets in mathematical programming. J. Math. Anal. Appl. **45**, 506–511 (1974)
36. Saint-Pierre, P.: Approximation of slow solutions to differential inclusions. Appl. Math. Optim. **22**, 311–330 (1990)
37. Schellenberger, J., Park, J., Conrad, T., Palsson, B.Ø.: BiGG: a biochemical genetic and genomic knowledgebase of large scale metabolic reconstructions. BMC Bioinform. **11**(1), 213 (2010)
38. Steuer, R.E.: Multiple Criteria Optimization: Theory, Computation, and Application. Wiley, New York (1986)
39. Tolsma, J., Barton, P.I.: DAEPACK: an open modeling environment for legacy models. Ind. Eng. Chem. Res. **39**(6), 1826–1839 (2000)
40. Wang, Z., Wu, X.: Piecewise integration of differential variational inequality. In: Simos, T., Psihoyios, G., Tsitouras, C. (eds.) Numerical Analysis and Applied Mathematics, vols. 1 and 2, vol. 2, pp. 912–915. AIP, Greece (2009)