Simulation and optimization of dynamic flux balance analysis models using an interior point method reformulation

Felipe Scott[a], Pamela Wilson[b], Raúl Conejeros[c], Vassilios S. Vassiliadis[d,*]

[a] *Green Technology Research Group, Facultad de Ingeniería y Ciencias Aplicadas, Universidad de los Andes, Chile, Mons. Álvaro del Portillo 12455, Las Condes, Santiago, 7620001, Chile*
[b] *Escuela de Ingeniería Industrial, Pontificia Universidad Católica de Valparaíso, Av. Brasil 2241, Valparaíso, 2362807, Chile*
[c] *Escuela de Ingeniería Bioquímica, Pontificia Universidad Católica de Valparaíso, Av. Brasil 2085, Valparaíso, 2362803, Chile*
[d] *Department of Chemical Engineering and Biotechnology, University of Cambridge, Cambridge CB3 0AS, UK*

Abstract

This work presents a novel, differentiable, way of solving dynamic Flux Balance Analysis (dFBA) problems by embedding flux balance analysis of metabolic network models within lumped bulk kinetics for biochemical processes. The proposed methodology utilizes transformation of the bounds of the embedded linear programming problem of flux balance analysis via a logarithmic barrier (interior point) approach. By exploiting the first-order optimality conditions of the interior-point problem, and with further transformations, the approach results in a system of implicit ordinary differential equations. Results from four case studies, show that the CPU and wall-times obtained using the proposed method are competitive with existing state-of-the art approaches for solving dFBA simulations, for problem sizes up to genome-scale. The differentiability of the proposed approach allows, using existing commercial packages, its application to the optimal control of dFBA problems at a genome-scale size, thus outperforming existing formulations as shown by two dynamic optimization case studies.

Keywords: Dynamic flux balance analysis, Ordinary differential equations with embedded optimization, Linear programming, Genome-scale metabolic network

1. Introduction

Genome-scale metabolic models provide a reliable representation of metabolism based on available information of cellular systems [18]. These models enable the mathematical representation of the metabolic processes occurring within the organism and may be analyzed further using available toolboxes based on mathematical optimization methods [42].

The central optimization task in metabolic networks is flux balance analysis (FBA; Orth et al. [27], Savinell and Palsson [33]). The most attractive feature of FBA is its ability to make quantitative predictions about a metabolic network without any need for detailed kinetic descriptions and given only the stoichiometry of the reactions, thus the number of published reconstructed genome-scale metabolic models has increased rapidly in recent years [40]. The only necessary inputs for FBA are the metabolic model (i.e., the network stoichiometry), a biologically meaningful objective and the growth and environmental conditions defining the substrates uptake rates. The fundamental assumption underlying FBA is that the system is at steady-state. The steady-state mass balance equation for each metabolite and environmental and growth conditions are mathematically described in the form of constraints for the optimization problem. Given that the system of equations describing the steady-state mass balances is under-determined (i.e., more reactions than metabolites exist), an infinite feasible solution set exists. To obtain a solution, a maximization principle is used as a surrogate for the true (and always unknown) totality of interactions. Typically this objective function is the maximization of the flux through the biomass formation reaction [27]. This results in a linear programming

*Corresponding author
Email address: vsv20@cam.ac.uk (Vassilios S. Vassiliadis)

(LP) formulation that can be solved readily using existing tools, such as GAMS or MATLAB™, or metabolic modeling frameworks such as the constrained-based modeling and analysis (COBRA) toolbox [34].

Dynamic Flux Balance Analysis (dFBA) is an extension of FBA enabling the simulation of the cellular dynamics of a culture system by assuming that cells reach an intracellular steady state rapidly in response to changes in the extracellular environment. In this way, the rates of product and biomass formation predicted by FBA are used to update the extracellular concentration in the environment. In turn, the changes in the environment produce variations in the uptake rates of substrates required for growth and metabolites production. In this way, the kinetics of the extracellular concentrations of substrates and products, often modeled as Ordinary Differential Equations (ODEs), are coupled to an FBA model, i.e, an LP problem. Several strategies have been developed to simulate dFBA models; and have been classified by Höffner et al. [15], who also offer a list of applications, as the Static Optimization Approach (SOA), the dynamic Optimization Approach (DOA) and the Direct Approach (DA).

The SOA approach uses the forward Euler's method to integrate the upper level ODE system and at each time step the embedded LP problem is solved using a suitable solver. As recognized by Gomez et al. [11], since most dFBA models are stiff, small time steps are required to ensure convergence, thus a large number of LP problems need to be solved to calculate the trajectory of the system, making this approach computationally expensive. The DOA approach is an attempt to use collocation methods to avoid the embedded nature of the dFBA problem. This approach discretizes the time horizon and transforms the problem to a nonlinear programming (NLP) problem. Mahadevan et al. [22] analyzed a network of 54 metabolites and 85 reactions using both the SOA and DOA approaches, concluding that the large number of constraints and variables introduced in the DOA approach limits its applicability to larger metabolic networks.

The DA approach includes the LP solver in the right hand side evaluator for the ordinary differential equations. Although this requires obtaining a solution of the LP problem at every evaluation of the right hand side, this approach can be implemented within implicit ODE integrators with adaptive step size for error control, thus reducing the number of integration steps compared to the use of SOA. In this regard, Gomez et al. [11] presented a DA implementation in MATLAB™, DFBAlab, that incorporates an LP feasibility problem and lexicographic linear optimization problems to generate an extended dynamic system for which the LP always has a solution. Lexicographic optimization augments the original LP problem, where typically the specific growth rate is maximized, by adding constraints from a user-predefined list of fluxes to deal with the (possible) existence of multiple flux distributions resulting in the same objective function. In this way, the LP problem is first solved by optimizing, for example, the specific growth rate. Next, a constraint is added specifying that the biomass flux should be equal or higher than the obtained optimum value and the next objective function in the predefined list is used. This idea has been used recently by Harwood et al. [12] and extended by exploiting the fact that, during an integration period, the optimal basis of the LP could remain unchanged thus transforming the dFBA problem in a system of semi-explicit index-1 differential algebraic equations. They also devised methods for detecting a change in the optimal basis of the LP and to update it. In this way, obtaining the solution of a dFBA simulation problem reduces to the integration of a semi-explicit index-1 system of equations until a change in the basis is detected. After updating the optimal basis, the integration can continue until the end of the integration horizon is attained. Theoretically this is the most elegant way for solving embedded LP problems within an ODE system and provides the most accurate solution without any approximation error. However, the active set method proposed by Höffner et al. [15] and Harwood et al. [12], is entirely equivalent to a basis identification method, such as the one implemented in DFBAlab. As such, it leads to a dynamic simulation that requires continuous monitoring and identification of any active set changes. This in turn constitutes a dynamic simulation involving discrete events (hybrid system).

Finally, Zhao et al. [41] propose a solution approach for dFBA problems with nonlinear objective functions, such as the maximization of the biomass yield or the maximization of the ATP yield per flux unit. In this approach, the Karush-Kuhn-Tucker (KKT) conditions of the LP are embedded resulting in a quasi differential-algebraic system of equations. Since the active set may change during the simulation, they use an extreme-ray-based transformation to update the active set. The largest problem solved using this approach consists of 45 intracellular reactions and took nearly 20 seconds. Considering that genome-scale models include thousands of reactions, new methods for solving large dFBA problems are required.

The system of differential equations with an embedded linear optimization problem can be reformulated as an index-1 differential and algebraic (DAE) system of equations by using the KKT conditions of the

embedded LP. However, the KKT conditions involve complementarity constraints. Complementarity is a relationship between variables where at least one of the variables must be at its bound [2], see Eqs. (9e) to (9h) for a typical set of complementarity constraints. These constraints are linearly dependent, which within the context of the aforementioned DAE system renders it unsolvable as it will have a linearly dependent Jacobian, as noticed by Zhao et al. [41].

Increasing demands for the sustainable and economically optimized synthesis of bioproducts, energy requirements and environmental concerns and demands for microbial strains that can produce valuable biochemicals led to efforts to improve the yield and productivity of fermentation processes by optimizing batch or fed-batch operation of bioreactors. Optimal control and parameter estimation applications of genome-scale dFBA models have been severely limited due to the computational burden of embedding a dFBA model into an optimal control problem for large models. In this regard, the solution of the bilevel optimization problem has been approached by resorting to its reformulation as a mathematical program with complementarity constraints (MPCC) [2]. In this approach the optimality conditions of the inner (FBA) optimization problem are imposed as constraints on the outer problem and the differential equations are discretized using different collocation strategies [4] leading to a NLP problem. This approach requires handling the complementary constraints by one of several regularization techniques to avoid the non-uniqueness of the constraint multipliers. Regularization approaches include the relaxation of the right hand side of the complementary constraint by a small positive value whose value decreases as the optimization proceeds, or the inclusion of a penalized sum of the complementary constraints in the objective function (see Section 3 in Baumrucker et al. [2]). Finally, MPCC solvers have been developed such as CONOPT-C [30] and also automatic reformulation tools of MPCCs are available such as the NLPEC meta-solver in GAMS, which allows using standard NLP solvers in GAMS.

The MPCC reformulation approach for solving optimization problems with embedded dFBA problems has been previously reported in the literature. Hjersted and Henson [13] studied the fed-batch optimization of a bioreactor with a small-scale model of *S. cerevisiae* metabolism. The approach used was to discretize the state variables, to model the feed stream as piecewise constant control inputs in time and to replace the LP problem by its KKT conditions, with this resulting in a nonlinear problem whose solution is limited by the size of the network. One year later, Hjersted et al. [14] presented a genome-scale analysis of the production of ethanol by *S. cerevisiae* in fed-batch culture, where in this work no attempts were made to optimize the performance of the fed-batch culture using optimal control, presumably because the metabolic model was too large to be handled by the current solution methods.

Kaplan et al. [17] proposed a parameter estimation formulation using a dFBA model of a yeast (42 metabolites and 48 reactions) handling the complementary constraints obtained by including the KKT conditions of the inner LP by using a Fischer-Burmeister smoothing function [2]. Raghunathan et al. [31] used variational inequalities to model switches in the objective function and in the uptake rates of substrates in a dFBA model of *S. cerevisiae* with 39 reactions. The model was embedded in a parameter estimation problem aimed to obtain the biomass composition in terms of macromolecular fractions of proteins, carbohydrates, nucleic acids and lipids. The optimization problem was reformulated as an MPCC and solved using CONOPT-C. Although the metabolic network analyzed was small, the resulting MPCC contains 33066 variables and 26192 constraints. Recently, Emenike et al. [9] applied a similar MPCC-collocation approach to the *in-silico* optimization of the production of recombinant proteins in *Pichia pastoris*. The metabolic network consists of 37 metabolites and 47 reactions, far from the available genome-scale metabolic models *P. pastoris*, such as *iPP*668, composed of 1.361 reactions and 1.177 metabolites [5].

Thereby, new methods are required in dFBA so as to be able to address three key points; (a) produce a differentiable simulation of dBFA so that it can be embedded in an optimal control solver, (b) be faster computationally than existing methods and able to handle genome-scale metabolic networks, and, (c) be able to deal with non-linear objective functions, also under the proviso that the weak Slater's condition [3] is satisfied for the resulting non-linear FBA problem (so that strong duality holds, as is the case with the LP formulation in this work).

In this work, a new approach for the solution of dFBA models is presented. The method relies on a transformation of the dFBA model to an implicit system of ordinary differential equations. This transformation is accomplished by using a logarithmic barrier approach (an Interior Point approach) for the inner LP problem. This approach is advantageous since it does not require the detection of a feasible set or an optimal basis, neither requires the repeated solution of LP problems. Moreover, our approach can be applied directly to

solve dynamic optimization problems with an embedded dFBA model. Hence, all three points presented in the previous paragraph are addressed with the contributions put forward with our present work, with actual implementation of non-linear objectives being a case that will be addressed in a future publication.

This paper is organized as follows: section 2 introduced the FBA and dFBA models as well as some useful properties of the interior point methods for the solution of LP problems, section 3 presents the interior point based formulation to solve dFBA models as implicit systems of ODEs. Finally, section 4 presents seven examples covering the simulation and dynamic optimization of dFBA models.

2. Theoretical background and problem formulation

2.1. Dynamic Flux Balance Analysis, dFBA

In the context of Flux Balance Analysis models, a mass balance for every identified metabolite is used to derive the stoichiometry of a set of the biochemical reactions taking place inside a cell and the transport of metabolites across the cell membrane, resulting in a set of linear equations. Let $\mathbf{v}$ be a vector of n fluxes, formed by each reaction rate expressed in mmol per hour per gram of dry biomass (mmol(gDWh)$^{-1}$). The flux of biomass is expressed as the specific growth rate (gDW(gDWh)$^{-1}$) to match the units of the experimental measurements. In FBA it is assumed that the rate of the internal reactions and transport rates of the $\widetilde{m}$ metabolites are faster when compared to the dynamics of the fermentation, resulting in a quasi-steady state mass balance [37] that can be expressed as $\mathbf{N}\mathbf{v} = 0$, where $\mathbf{N}$ is an $\widetilde{m} \times n$ matrix with $\text{rank}(\mathbf{N}) = \widetilde{m}$.

Generally, the resulting system of linear equations cannot be solved as the number of variables is larger than the number of equations. Thereby, it is assumed that a cellular objective exists, such as the maximization of the specific cell growth under the prevailing external conditions. Hence, the following linear programming (LP) problem can be formulated:

$$\begin{aligned} \min_{\mathbf{v}} \quad & -\mathbf{c}^{\mathbf{T}}\mathbf{v} \\ \text{s.t.} \quad & \mathbf{N}\mathbf{v} = 0, \\ & \mathbf{v}^{lo} \leq \mathbf{v} \leq \mathbf{v}^{up}, \end{aligned} \tag{1}$$

where, $\mathbf{c}$ is a column vector of length n with positive or zero entries and $\mathbf{v}^{lo}$ and $\mathbf{v}^{up}$ are bounds on the optimization variable $\mathbf{v}$ representing the uptake and product fluxes of the problem. The bounds are chosen so as to restrict the value of the fluxes to within realistic intervals.

FBA is a conveniently simple way to incorporate biochemical pathway information without the need of intracellular kinetics to any bulk phase, macroscopic model of biochemical processes. The equality and inequality constraints of the FBA problem form a polytope where the problem is feasible. The optimal solutions of the LP problem can lay on a vertex of the polytope, and be unique, or be non-unique solutions if the objective function hyperplane is parallel to a facet of the constraint polytope at the solution. This is a mathematical shortcoming of the model which fails to produce a uniquely defined set of fluxes for the underlying biochemical reaction network, without any further specialized manipulation for the case of FBA, such as lexicographic optimization [12].

A second shortcoming of FBA is that it may highlight parts of the biochemical reaction pathway as active when in reality there is no way of being certain regarding their activity without further experimental confirmation. This is the result of the problem being incomplete in terms of having more variables than equations to obtain a solution which defines the state of the network (the full set of independent fluxes). This second problem is an inevitability regardless of the method chosen to solve the FBA LP problem.

In a Dynamic Flux Balance Analysis, the mass balance of the measured species in the bioreactor media is defined by a system of ordinary differential equations (ODEs) accounting for the variation of their concentrations in time. The consumption of substrates from the media is linked to the uptake substrate fluxes in the FBA model. This is represented by a set of algebraic equations. Thus, for given substrate uptake rates a solution of the FBA problem (an optimal flux distribution) can be obtained, which in turn results in a specific growth rate and specific product rates. These rates will modify the concentrations of substrates, biomass and products in the culture media.

The FBA problem (Eq. 1) will be modified to accommodate the specification of the substrate uptake rates. First, let the matrix $\mathbf{A} \in \mathbb{R}^{m \times n}$, with $m = \widetilde{m} + p$, be defined as:

$$\mathbf{A} = \begin{bmatrix} \mathbf{N} \\ \mathbf{C} \end{bmatrix}, \tag{2}$$

where $\mathbf{C} \in \mathbb{R}^{p \times n}$ is a matrix containing only zeros and ones, so that the product $\mathbf{Cv} \in \mathbb{R}^p$ denotes all p substrate uptake rates. Let be $\mathbf{q}_{\text{upt}}(\mathbf{x})$ a vector of p uptake rates defined by algebraic functions of the species concentrations in the media ($\mathbf{x}$), then for a given value of the concentrations of the species in the media, the mass balance of the intracellular species can be written as:

$$\mathbf{Av} = \mathbf{b} = \begin{bmatrix} 0 \\ \mathbf{q}_{\text{upt}}(x) \end{bmatrix}, \tag{3}$$

The aforementioned situation is described by the following system of ordinary differential equations with an embedded linear programming problem:

$$\begin{aligned} &\frac{d\mathbf{x}(t)}{dt} = f(\mathbf{x}(\mathbf{t}), \mathbf{v}(t)), \;\; \mathbf{x}(0) = \mathbf{x}_0, \\ &\mathbf{q}_{\text{upt}}(t) = \mathbf{g}(\mathbf{x}(t)), \\ &\mathbf{v}(t) \in \; \arg\min_{\mathbf{w}} \{-\mathbf{c}^{\mathbf{T}}\mathbf{w} | \mathbf{Aw} = \mathbf{b}(t) = [0 \quad \mathbf{q}_{\text{upt}}(t)]^T, \mathbf{w}^{lo} \leq \mathbf{w} \leq \mathbf{w}^{up}\}, \end{aligned} \tag{4}$$

where $\mathbf{x} \in \mathbb{R}^d$ corresponds to time dependent concentration whose evolution is controlled by a continuous function $f : \mathbb{R}^d \times \mathbb{R}^n \to \mathbb{R}^d$, $\mathbf{q}_{\text{upt}} \in \mathbb{R}^p$ represents the specific uptake rates of p substrates and $\mathbf{g} : \mathbb{R}^d \to \mathbb{R}^p$ is a continuous C^1 vector function. Finally, vector $\mathbf{v}(t)$ corresponds to the flux vector minimizing the FBA problem. It is noted that strictly speaking the solution of the embedded LP may be an infinite set of values, rather than a singular vector, achieving the same objective function in the case where the objective function hyperplane of the LP is parallel to a facet of its feasible polytope. This set is defined by the linear combination of the vertices of the active facet of the polytope.

2.1.1. Approaches for solving linear programming problems (LP)

Several approaches exist for handling the linear equality and inequality constraints in the LP problem (Eq. 4). A linear programming problem can be written as a primal problem and its corresponding dual problem. Alternatively, an augmented objective function can be written by adding a penalization of the inequality constraints (the bounds on fluxes). In this section, we will show that the latter results in a differentiable set of equations that replaces the optimization problem, and thus can be used to transform Eq. 4 from a ODE system with an embedded LP into a set of differential and algebraic equations (DAE).

Moreover, we will show that the Karush-Kuhn-Tucker (KKT) conditions of an LP reformulation of the embedded FBA problem via a primal-dual log-barrier transformation results in a fully differentiable model, comprised purely of algebraic equations whose solution depends *parametrically* on the extracellular environment state variables (metabolite and substrate concentrations, pH, etc.). Formulating the FBA LP problem KKT conditions directly results in MPCC, thus requiring specialized solution techniques and collocation of the differential equations. Use of the interior point formulation handles all these issues automatically and smoothly.

The differentiability of the transformed dFBA model via the primal-dual approach is paramount in order to generate reliably sensitivity equations, as discussed in section 3 and Problem 5 of this article. Although interior point methods become more efficient when dealing with large- to huge-size LP problems, it should be stressed that here we do not have a free-standing LP problem: the LP problem is embedded within an ODE bulk-phase model of bioreactors. As evidenced by the computational results, our approach using standard, state-of-the-art, dynamic process simulators results in highly competitive solution times even when compared with the customized DFBAlab tool [11].

Although the proposed approach is within the category of complementarity conditions relaxations, as the MPCC regularization methodology reviewed in the Introduction section by being effectively a μ-relaxation of the complementarity conditions of the LP associated with FBA, it results from writing the KKT conditions of optimality of the LP transformed via the interior point method. As such, it has the property that it always results in a unique solution regardless of whether or not the exact solution of the LP lies at the vertex or a facet of the constrained polytope, as in the case where the objective function hyperplane is parallel to a facet of the polytope at the solution. By the properties of interior point methods in the former case the

solution will be strictly interior in the vicinity of the active vertex, or in the latter case again in the interior of the polytope in the vicinity of the analytic center of the active facet [19]. It is noted that such a solution is convenient for simulation and optimization purposes, as one does not have to worry about its uniqueness. Arguably, this constitutes an arbitrary choice from among the possible active vertices in this case, which lexicographic optimization approaches can handle provided that a suitable ordering of uptake and product fluxes is possible to define *a priori* [12].

An important property of the μ-relaxation which results from the consideration of an interior point transformation is that the bounds in the variables and consequently on the associated Lagrange multipliers of the bounds are always satisfied strictly. Furthermore, the μ-relaxation of complementarity suffices for the solution of the associated embedded LP problem without the need for any further introduction of transformations.

In attempting to solve computationally the semi-explicit index one DAE system given in equations (16a to 16f) it is observed that numerical solvers exhibit difficulty to converge the initialization phase. This occurs because these numerical solvers and simulation packages employed a general purpose Newton method. However, we are solving non-linear systems which arise from complementarity conditions of an interior point reformulation of the LP, and such applications require a customized Newton solver which retracts the search space to be strictly within the bounds of the variables. To alleviate this problem an index reduction of the DAE system is applied to render it into a pure implicit ODE system, for which consistent initial conditions for all states can be provided conveniently outside the integration phase through an interior-point LP solver. The particular choice of μ-relaxations of the complementarity conditions results in a very straightforward coupled linear ODE subsystem as it will be shown below.

Finally a significant difference of our implementation over the work presented in Raghunathan et al. [31], Hjersted and Henson [13], Kaplan et al. [17] and Emenike et al. [9] is that right at the outset we are aiming for very large scale dynamic models (genome scale) which can include several thousands of reactions and metabolites, and to be implementable as part of larger flowsheets, both for simulation, optimization, and parameter estimation purposes, through implementation in existing advanced equation oriented flowsheeting packages such as gPROMS [29].

In the remaining of this section we follow closely the approach presented by Monteiro and Adler [24], while similar treatments can be found in the work of Kojima et al. [19] and Megiddo [23]. We start by defining the following pair of primal $\mathcal{P}$ and dual $\mathcal{D}$ problems for the inner LP in problem 4:

Primal problem $\mathcal{P}$:

$$\begin{aligned} &\min_{\mathbf{v}} -\mathbf{c}^T\mathbf{v} \\ \text{s.t.} \quad & \\ &\mathbf{A}\mathbf{v} = \mathbf{b}, \\ &\mathbf{v}^{up} - \mathbf{v} \geq 0, \\ &\mathbf{v} - \mathbf{v}^{lo} \geq 0. \end{aligned} \tag{5}$$

Dual problem $\mathcal{D}$:

$$\begin{aligned} &\max_{\boldsymbol{\lambda},\,\mathbf{z},\,\mathbf{y}} -\mathbf{b}^T\boldsymbol{\lambda} + \mathbf{y}^T\mathbf{v}^{lo} - \mathbf{z}^T\mathbf{v}^{up} \\ \text{s.t.} \quad & \\ &\mathbf{A}^T\boldsymbol{\lambda} - \mathbf{y} + \mathbf{z} = \mathbf{c}, \\ &\mathbf{z} \geq 0, \\ &\mathbf{y} \geq 0, \end{aligned} \tag{6}$$

where $\boldsymbol{\lambda} \in \mathbb{R}^m$, $\mathbf{y}$ and $\mathbf{z}$ both in $\mathbb{R}^n$, are the dual variables of problem $\mathcal{P}$.

In our work, we explore the application of logarithmic barrier functions to handle the bounds on variables in problem $\mathcal{P}$, allowing its reformulation as:

Problem $\mathcal{P}_\mu$:

$$\begin{aligned} \min_{\mathbf{v}} & -\mathbf{c}^T\mathbf{v} - \mu \sum_{i=1}^{n} \left[\ln\left(v_i - {v_i}^{lo}\right) + \ln\left(v_i^{up} - v_i\right)\right] \\ \text{s.t.} & \\ & \mathbf{A}\mathbf{v} = \mathbf{b}, \end{aligned} \tag{7}$$

where $\mu > 0$ is the barrier penalty parameter. Before analyzing if problems $\mathcal{P}_\mu$, $\mathcal{P}$ and $\mathcal{D}$ are equivalent, the following assumptions are imposed [24]. These assumptions are required to guarantee the existence of a non empty solution space for the embedded LP.

Assumption 1. *The problems $\mathcal{P}_\mu$, $\mathcal{P}$ and $\mathcal{D}$ have the following properties (an adaptation of the properties stated in Monteiro and Adler [24]):*

a. *The set $S \equiv \{\mathbf{v} \in \mathbb{R}^n;\ \mathbf{A}\mathbf{v} = \mathbf{b},\ \mathbf{v} - \mathbf{v}^{lo} \geq 0,\ \mathbf{v}^{up} - \mathbf{v} \geq 0\}$ is non-empty for $\mathbf{b} = \mathbf{b}(t)$ for all t in the integration time-span $[t_0, t_f]$.*
b. *The set $T \equiv \{(\boldsymbol{\lambda}, \mathbf{y}, \mathbf{z}) \in \mathbb{R}^{m+n+n};\ \mathbf{A}^T\boldsymbol{\lambda} - \mathbf{y} + \mathbf{z} = \mathbf{c},\ \mathbf{y} \geq 0,\ \mathbf{z} \geq 0\}$ is non-empty.*
c. $\text{rank}(\mathbf{A}) = m = \text{rank}(\mathbf{N}) + \text{rank}(\mathbf{C}) = \tilde{m} + p$.

Thus, the sets S and T are interior feasible solutions of problems $\mathcal{P}$ and $\mathcal{D}$, respectively. Under Assumption 1, we will show that the first-order conditions for $\mathcal{P}$ and $\mathcal{D}$ are identical. First-order necessary optimality conditions (cf. Theorem 12.1 in Nocedal and Wright [25]) for problems $\mathcal{P}$ and $\mathcal{D}$ are obtained from its Lagrangian functions.

The Lagrangian function of problem $\mathcal{P}$ is:

$$L(\mathbf{v}, \boldsymbol{\lambda}, \mathbf{y}, \mathbf{z}) = -\mathbf{c}^T\mathbf{v} + \boldsymbol{\lambda}^T(\mathbf{A}\mathbf{v} - \mathbf{b}) - \mathbf{z}^T(\mathbf{v}^{up} - \mathbf{v}) - \mathbf{y}^T(\mathbf{v} - \mathbf{v}^{lo}). \tag{8}$$

Hence, the first-order necessary conditions for $\mathbf{v}^*$ to be a solution of $\mathcal{P}$ are that there exists vectors $\boldsymbol{\lambda}^*$, $\mathbf{y}^*$ and $\mathbf{z}^*$ such that:

$$-\mathbf{c} + \mathbf{A}^T\boldsymbol{\lambda} - \mathbf{y} + \mathbf{z} = 0, \tag{9a}$$
$$\mathbf{A}\mathbf{v} = \mathbf{b}, \tag{9b}$$
$$\mathbf{v}^{up} - \mathbf{v} \geq 0, \tag{9c}$$
$$\mathbf{v} - \mathbf{v}^{lo} \geq 0, \tag{9d}$$
$$\mathbf{y} \geq 0, \tag{9e}$$
$$\mathbf{z} \geq 0, \tag{9f}$$
$$z_i(v_i^{up} - v_i) = 0, \quad i = 1, 2..., n, \tag{9g}$$
$$y_i(v_i - v_i^{lo}) = 0, \quad i = 1, 2..., n, \tag{9h}$$

holds for $\mathbf{v} = \mathbf{v}^*$, $\boldsymbol{\lambda} = \boldsymbol{\lambda}^*$, $\mathbf{z} = \mathbf{z}^*$ and $\mathbf{y} = \mathbf{y}^*$.

For the dual problem $\mathcal{D}$, the Lagrangian function corresponds to:

$$\widetilde{L}(\boldsymbol{\lambda}, \mathbf{y}, \mathbf{z}, \mathbf{v}, \boldsymbol{\mu}_1, \boldsymbol{\mu}_2) = \mathbf{b}^T\boldsymbol{\lambda} - \mathbf{y}^T\mathbf{v}^{lo} + \mathbf{z}^T\mathbf{v}^{up} - \mathbf{v}^T(-\mathbf{c} + \mathbf{A}^T\boldsymbol{\lambda} - \mathbf{y} + \mathbf{z}) - \boldsymbol{\omega}_1^T\mathbf{z} - \boldsymbol{\omega}_2^T\mathbf{y}. \tag{10}$$

The vector triplet ($\boldsymbol{\lambda}^*$, $\mathbf{y}^*$, $\mathbf{z}^*$) will be a solution of the dual problem if there exists vectors $\mathbf{v}^*$, $\boldsymbol{\omega}_1^*$ and $\boldsymbol{\omega}_2^*$ in $\mathbb{R}^n$, multipliers for the equality, upper bounds and lower bounds in problem $\mathcal{D}$, respectively, such that the first order necessary conditions:

$$\frac{\partial \widetilde{L}}{\partial \boldsymbol{\lambda}} = \mathbf{A}\mathbf{v} - \mathbf{b} = 0 \tag{11a}$$

$$\frac{\partial \widetilde{L}}{\partial \mathbf{y}} = \mathbf{v} - \mathbf{v}^{lo} - \boldsymbol{\omega}_2 = 0 \tag{11b}$$

$$\frac{\partial \widetilde{L}}{\partial \mathbf{z}} = \mathbf{v}^{up} - \mathbf{v} - \boldsymbol{\omega}_1 = 0 \tag{11c}$$

$$\frac{\partial \widetilde{L}}{\partial \mathbf{v}} = \mathbf{A}^T\boldsymbol{\lambda} - \mathbf{y} + \mathbf{z} - \mathbf{c} = 0 \tag{11d}$$

$$\mathbf{y} \geq 0, \tag{11e}$$

$$\mathbf{z} \geq 0 \tag{11f}$$

$$\boldsymbol{\omega}_1 \geq 0 \tag{11g}$$

$$\boldsymbol{\omega}_2 \geq 0 \tag{11h}$$

$$\omega_{1i} z_i = 0 \tag{11i}$$

$$\omega_{2i} y_i = 0, \quad i = 1, 2..., n \tag{11j}$$

holds for $(\boldsymbol{\lambda}, \mathbf{y}, \mathbf{z}) = (\boldsymbol{\lambda}^*, \mathbf{y}^*, \mathbf{z}^*)$ and $(\mathbf{v}, \boldsymbol{\omega}_1, \boldsymbol{\omega}_2) = (\mathbf{v}^*, \mathbf{v^{up}} - \mathbf{v}^*, \mathbf{v}^* - \mathbf{v^{lo}})$. By replacing the optimal values for the multipliers in the system above, it can be verified that the first-order for $\mathcal{P}$ and $\mathcal{D}$ are identical.

Finally, the first-order necessary conditions for problem $\mathcal{P}_\mu$ are:

$$\mathbf{A}^T\boldsymbol{\lambda} - \mathbf{y} + \mathbf{z} - \mathbf{c} = 0, \tag{12a}$$

$$\mathbf{A}\mathbf{v} = \mathbf{b}, \tag{12b}$$

$$z_i(v_i^{up} - v_i) = \mu, \tag{12c}$$

$$y_i(v_i - v_i^{lo}) = \mu, \quad i = 1, 2..., n. \tag{12d}$$

Now, we apply the following proposition from Monteiro and Adler [24] to problem $\mathcal{P}_\mu$.

Proposition 2.1. *If Assumption 1.a holds and let $\mu > 0$ be given, then $\mathcal{P}_\mu$ has an optimal solution if and only if the set of optimal solutions of $\mathcal{P}$ is non-empty and bounded.*

A proof of proposition 2.1 is given in Megiddo [23]. As stated by Monteiro and Adler [24], this implies that if $\mathcal{P}_\mu$ has a solution for some $\mu > 0$, then it has a solution for all $\mu > 0$. Moreover, the Duality Theorem of Linear Programming states that if either problem $\mathcal{P}$ or $\mathcal{D}$ has a solution with finite optimal objective value, then so does the other, and the objective values are equal (Theorem 13.1 in Nocedal and Wright [25]). As a consequence the following corollary can be stated.

Corollary 2.1. *Under Assumption 1.a and 1.b, problem $\mathcal{P}_\mu$ has a unique solution $\mathbf{v}(\mu)$, $\boldsymbol{\lambda}(\mu)$, $\mathbf{y}(\mu)$ and $\mathbf{z}(\mu)$ for all $\mu > 0$.*

By analyzing the system of equations derived by applying the Karush-Kuhn-Tucker conditions to problem $\mathcal{P}$, it can concluded from the last two equations that if $\mathbf{v} \in S$ and $\mu > 0$, then $\mathbf{z} > 0$ and $\mathbf{y} > 0$. The first equation implies that $(\boldsymbol{\lambda}, \mathbf{y}, \mathbf{z})$ is an interior feasible solution to the dual problem $\mathcal{D}$. From assumption 1.c, it can be concluded that there is a unique $\boldsymbol{\lambda}$ satisfying Eqs. (12a) to (12d).

Finally, the following proposition ensures that the solution of $\mathcal{P}_\mu$ is identical to the solution of $\mathcal{P}$ and $\mathcal{D}$ as $\mu \to 0$.

Proposition 2.2. *If Assumption 1 holds, as $\mu \to 0$, $\mathbf{v}(\mu)$ and $(\boldsymbol{\lambda}(\mu), \mathbf{y}(\mu), \mathbf{z}(\mu))$ converges to the optimal solutions of problems $\mathcal{P}$ and $\mathcal{D}$ respectively.*

Proof. Let $\mathbf{w}(\mu) = (\mathbf{v}(\mu), \boldsymbol{\lambda}(u), \mathbf{y}(\mu), \mathbf{z}(\mu))$ be the point satisfying Eqs. (12a) to (12d). The duality gap at this point is by definition:

$$g(\mathbf{w}) = -\mathbf{c}^T\mathbf{v} - (-\mathbf{b}^T\boldsymbol{\lambda} + \mathbf{y}^T\mathbf{v}^{lo} - \mathbf{z}^T\mathbf{v}^{up}). \tag{13}$$

Using Eqs. (12a) and (12b), one can show that:

$$g(\mathbf{w}) = \sum_{i=1}^{n} \left[z_i(v_i^{up} - v_i) + y_i(v_i - v_i^{lo}) \right]. \tag{14}$$

By using Eqs. (12c) and (12d) it can be concluded that:

$$g(\mathbf{w}) = 2n\mu. \tag{15}$$

Therefore, the duality gap converges to zero as $\mu \to 0$, implying that the objective functions of problems $\mathcal{P}$ and $\mathcal{D}$ converge to a common optimal value. □

3. An interior point based formulation for ODEs with embedded LPs

The system of ordinary differential equations with en embedded LP defined by Eq. (4) can be transformed to a differential and algebraic equation system (DAEs) by replacing the embedded LP with the first-order necessary conditions of optimality for problem $\mathcal{P}_\mu$ (Eqs. (12a) to (12d)):

$$\frac{d\mathbf{x}(t)}{dt} = f(\mathbf{x}(t), \mathbf{v}(t)), \quad \mathbf{x}(0) = \mathbf{x}_0, \tag{16a}$$

$$\mathbf{q}_{\text{upt}}(t) = g(\mathbf{x}(t)), \tag{16b}$$

$$\mathbf{A}\mathbf{v}(t) = \mathbf{b}(t) = \begin{bmatrix} 0 \\ \mathbf{q}_{\text{upt}}(\mathbf{x}(t)) \end{bmatrix}, \tag{16c}$$

$$z_i(t)(v_i^{up} - v_i(t)) = \mu, \tag{16d}$$

$$y_i(t)(v_i(t) - v_i^{lo}) = \mu, \quad i = 1, 2..., n. \tag{16e}$$

$$-\mathbf{A}^T \boldsymbol{\lambda}(t) + \mathbf{y}(t) - \mathbf{z}(t) = -\mathbf{c}. \tag{16f}$$

It is noted that in the above model, $v(t)$ constitutes the unique solution of the reformulated instantaneous LP problem by the properties of interior point methods as outlined in section 2.1.1.

The inclusion of the Karush-Kuhn-Tucker conditions of a primal-dual formulation for an LP problem contains a μ relaxation of the complementarity conditions of the LP (bounds values times the corresponding Lagrange multipliers, c.f. Eq. (7)) in a smooth way which allows its continuous integration without requiring further regularization. This is equivalent, in effect, to the regularization scheme presented in Baumrucker et al. [2] in equations (19a)-(19e), noting that no inequality constraints are left in the formulation which allows their continuous and uninterrupted integration into a uniform DAE/ODE system. A similar approach, to handle embedded LP problems into DAE models is presented in Kaplan et al. [17], but it is worth noting that in equations (14) and (15) of their paper the approach adopted is non-differentiable in the right hand-side and hence would not be suitable for smooth integration with the aim to generate sensitivity equations.

Unless explicitly stated, the time dependencies of the variables will be omitted for simplicity. The DAE (Eqs. 16a to 16f) can be transformed to a system of implicit ODEs by differentiation to yield:

$$\frac{d\mathbf{x}}{dt} = f(\mathbf{x}, \mathbf{v}), \quad x(0) = x_0, \tag{17a}$$

$$\frac{d\mathbf{q}_{\text{upt}}}{dt} = \frac{dg(\mathbf{x})}{d\mathbf{x}} \frac{d\mathbf{x}}{dt}, \tag{17b}$$

$$\frac{dz_i}{dt} = \frac{\mu}{(v_i^{up} - v_i)^2} \frac{dv_i}{dt}, \quad i = 1, 2..., n, \tag{17c}$$

$$\frac{dy_i}{dt} = -\frac{\mu}{(v_i - v_i^{lo})^2}\frac{dv_i}{dt}, \quad i = 1, 2..., n, \tag{17d}$$

$$\mathbf{A}\frac{d\mathbf{v}}{dt} = \frac{d\mathbf{b}}{dt} = \begin{bmatrix} 0 \\ \frac{d\mathbf{q}_{\text{upt}}}{dt} \end{bmatrix}, \tag{17e}$$

$$-\mathbf{A}^T\frac{d\boldsymbol{\lambda}}{dt} + \frac{d\mathbf{y}}{dt} - \frac{d\mathbf{z}}{dt} = 0. \tag{17f}$$

The initial point must correspond to the solution of the primal-dual barrier formulation of the FBA problem at the initial time ($t_0 = 0$), derived with the same barrier parameter (μ) that is used in the simulation of the dFBA model considered.

Finally, the implicit system of equations can be further reduced (Reduced - implicit ODE or R-iODE). This is performed not only to decrease the number of differential variables during the numerical integration, but also to create a representation of the system of differential equations suitable to be used in the proofs and propositions in this section:

R-iODE problem:

$$\frac{d\mathbf{x}}{dt} = f(\mathbf{x}, \mathbf{v}), \quad x(0) = x_0 \tag{18a}$$

$$\frac{d\mathbf{q}_{\text{upt}}}{dt} = \frac{dg(\mathbf{x})}{d\mathbf{x}}\frac{d\mathbf{x}}{dt}, \tag{18b}$$

$$\widetilde{\mathbf{A}} = \begin{bmatrix} \mathbf{A} & \mathbf{0} \\ \mathbf{D}(\mu, \mathbf{v}) & \mathbf{A}^T \end{bmatrix} \begin{bmatrix} \frac{d\mathbf{v}}{dt} \\ \frac{d\boldsymbol{\lambda}}{dt} \end{bmatrix} = \begin{bmatrix} \frac{d\mathbf{b}}{dt} \\ \mathbf{0} \end{bmatrix}, \tag{18c}$$

where $\mathbf{D}(\mu, \mathbf{v})$ is a positive definite diagonal matrix whose entries are defined as:

$$D_{i,i}(\mu, \mathbf{v}) = \frac{\mu}{(v_i^{\text{up}} - v_i)^2} + \frac{\mu}{(v_i - v_i^{\text{lo}})^2} \tag{19}$$

Provided that Assumption 1 holds, then the inner linear problem $\mathcal{P}$ in Eq. 4 has a non-empty and bounded feasible region, the following result can be stated:

Proposition 3.1. *Under assumption 1, the inner linear program in equation 4 has a unique solution* $\mathbf{v}^*$*, with* $\mathbf{v}^{lo} < \mathbf{v}^* < \mathbf{v}^{up}$ *for a given* $\mu > 0$ *at a certain time* $t \in [t_0, t_f]$*. Then, the system of equations (18a) to (18c) has a unique solution for* $\frac{d\mathbf{v}}{dt}$*,* $\frac{d\boldsymbol{\lambda}}{dt}$*,* $\frac{d\mathbf{q}_{\mathbf{upt}}}{dt}$ *and* $\frac{d\mathbf{x}}{dt}$ *at* t*.*

Proof. Under the assumptions of Proposition 3.1, the diagonal matrix $\mathbf{D}(\mu, \mathbf{v})$ and its inverse always exist. Using block matrices inversion [20], the inverse of $\widetilde{\mathbf{A}}$ is:

$$\widetilde{\mathbf{A}}^{-\mathbf{1}} = \begin{bmatrix} \mathbf{D}^{-1}\mathbf{A}^T(\mathbf{A}\mathbf{D}^{-1}\mathbf{A}^T)^{-1} & \mathbf{D}^{-\mathbf{1}} - \mathbf{D}^{-1}\mathbf{A}^T(\mathbf{A}\mathbf{D}^{-1}\mathbf{A}^T)^{-1}\mathbf{A}\mathbf{D}^{-1} \\ -(\mathbf{A}\mathbf{D}^{-1}\mathbf{A}^T)^{-1} & (\mathbf{A}\mathbf{D}^{-1}\mathbf{A}^T)^{-1}\mathbf{A}\mathbf{D}^{-1} \end{bmatrix} \tag{20}$$

and the solution can be expressed in terms of the inverse of the Schur complement $\mathbf{A}\mathbf{D}^{-1}\mathbf{A}^T$, which is well defined since $\mathbf{A}$ is of full rank [23]. Thereby, $\frac{d\mathbf{v}}{dt}$ and $\frac{d\boldsymbol{\lambda}}{dt}$ can be calculated as the product $\widetilde{\mathbf{A}}^{-1}\begin{bmatrix} \frac{d\mathbf{b}}{dt} \\ \mathbf{0} \end{bmatrix}$, which is unique, then $\frac{d\mathbf{q}_{\text{upt}}}{dt}$ and $\frac{d\mathbf{x}}{dt}$ are also unique at t. □

Consider an explicit integration scheme, such as the Euler's explicit method, applied to the integration of Eqs. (18a) to (18c). At time $t = 0$, $x(0) = x_0$, $\mathbf{q}_{\text{upt}}(0) = g(\mathbf{x}(t_0))$ and the solution of the inner LP problem can be obtained by solving Eqs. 12a to 12d by using an interior-point based algorithm to yield a point $\mathbf{v}(\mu, \mathbf{q}_{\text{upt}}(0))$, $\boldsymbol{\lambda}(\mu, \mathbf{q}_{\text{upt}}(0))$, $\mathbf{y}(\mu, \mathbf{q}_{\text{upt}}(0))$ and $\mathbf{z}(\mu, \mathbf{q}_{\text{upt}}(0))$. If these solutions exist, it must be true that $\mathbf{v^{lo}} < \mathbf{v}(\mu, \mathbf{q}_{\text{upt}}(0)) < \mathbf{v^{up}}$. Then, the next step in the discretized trajectory of the system of differential equations (18a) to (18c) is uniquely determined. These ideas are formalized in the following theorem:

Theorem 3.1. *Suppose assumption 1 holds for every inner LP problem along a time interval $t \in [t_0, t_f]$ and that a point $x(t_0) = x_0$, $\mathbf{q}_{\text{upt}}(t_0) = g(\mathbf{x}(0))$, $\mathbf{v}(\mu, \mathbf{q}_{\text{upt}}(t_0))$, $\boldsymbol{\lambda}(\mu, \mathbf{q}_{\text{upt}}(t_0))$, $\mathbf{y}(\mu, \mathbf{q}_{\text{upt}}(t_0))$ and $\mathbf{z}(\mu, \mathbf{q}_{\text{upt}}(t_0))$ can be calculated at $t = t_0$ satisfying the DAE system defined by Eqs. (16a) to (16f). If an explicit integration scheme is used to calculate a discretized trajectory (in time) of the system of ODEs defined by Eqs. (18a) to (18c), then this trajectory is unique.*

Proof. The proof will proceed by induction. First consider that a solution of the inner LP of the dFBA problem (Eq. 4) exists at time zero for an initial condition $\mathbf{x}(0) = \mathbf{x}_0$, thus yielding $\mathbf{v}_0$, $\mathbf{y}_0$, $\mathbf{z}_0$ and $\boldsymbol{\lambda}_0$ Also consider that the integration time horizon is discretized, using a constant step size s for simplicity, such that $t_{k+1} = t_k + s$. If the explicit Euler method is used to calculate the next step in time of the differential variables in the system of ODEs defined by Eqs. (18a) to (18c), its value will be given by:

$$\begin{aligned} \mathbf{x}(0+s) &= f(\mathbf{x}_0, \mathbf{v}_0)s, \\ \mathbf{q}_{\text{upt}}(0+s) &= \frac{dg(\mathbf{x}_0)}{d\mathbf{x}} f(\mathbf{x}_0, \mathbf{v}_0)s, \\ \begin{bmatrix} \mathbf{v}(0+s) \\ \boldsymbol{\lambda}(0+s) \end{bmatrix} &= s\left(\widetilde{\mathbf{A}}(\mu, \mathbf{v}_0)\right)^{-1} \begin{bmatrix} \frac{\mathbf{db}}{\mathbf{dt}}(0) \\ \mathbf{0} \end{bmatrix}. \end{aligned} \tag{21}$$

Similarly, the value of the differential variables at t_{k+1} can be calculated as:

$$\begin{aligned} \mathbf{x}(t_k+s) &= f(\mathbf{x}_k, \mathbf{v}_k)s, \\ \mathbf{q}_{\text{upt}}(t_k+s) &= \frac{dg(\mathbf{x}_k)}{d\mathbf{x}} f(\mathbf{x}_k, \mathbf{v}_k)s, \\ \begin{bmatrix} \mathbf{v}(t_k+s) \\ \boldsymbol{\lambda}(t_k+s) \end{bmatrix} &= s\left(\widetilde{\mathbf{A}}(\mu, \mathbf{v}_k)\right)^{-1} \begin{bmatrix} \frac{d\mathbf{b}}{dt}(t_k) \\ \mathbf{0} \end{bmatrix}. \end{aligned} \tag{22}$$

Since the right hand side of Eq. 22 is a function of the known values of $\mathbf{v}$ and $\mathbf{x}$ at t_k, then the values of the differential variables at t_{k+1} are uniquely determined by equations Eq. 22. □

Theorem 3.2. *Under the assumptions of Theorem 3.1 the value of the duality gap defined in Eq. (14) is constant for every value of $t \in [t_0, t_f]$ and depends only on the value of μ. Thus, as $\mu \to 0$, at each time t, the solution of Eqs. (18a) to (18c) $\mathbf{v}(\mu)$, $\boldsymbol{\lambda}(\mu)$ and $\mathbf{y}(\mu)$ and $\mathbf{z}(\mu)$ from Eqs. (17c) and (17d) converges to the optimal solutions of the embedded linear programming problem in Eq. (4).*

Proof. We start from the definition of the duality gap in Eq. (14) and differentiate it with respect to time to yield:

$$\frac{dg(\mathbf{w})}{dt} = (\mathbf{v}^{up} - \mathbf{v})^T \frac{d\mathbf{z}}{dt} - \mathbf{z}^T \frac{d\mathbf{x}}{dt} + (\mathbf{v} - \mathbf{v}^{lo})^T \frac{d\mathbf{y}}{dt} - \mathbf{y}^T \frac{d\mathbf{x}}{dt}. \tag{23}$$

Replacing $\frac{d\mathbf{z}}{dt}$ and $\frac{d\mathbf{y}}{dt}$ by their definitions from Eqs. (17c) and (17d), respectively, and factorizing we obtain:

$$\frac{dg(\mathbf{w})}{dt} = \left[\frac{\mu}{v_i^{up} - v_i} - \frac{\mu}{v_i - v_i^{lo}} - z_i + y_i\right] \frac{d\mathbf{x}}{dt}. \tag{24}$$

The term in brackets in the previous equation is zero (see Eqs. (17c) and (17d)), thereby:

$$\frac{dg(\mathbf{w})}{dt} = 0 \to g(t) = g(t_0) = 2n\mu \tag{25}$$

In practice, the value of the duality gap will also depend on the integration error tolerance, which in turn is controlled by the step size h of the integration method and its order, n:

$$g(t) = 2n\mu + \mathcal{O}(h^n) \tag{26}$$

□

3.1. Implementation

The Direct Approach (DA) for the solution of the ODEs with embedded LPs was implemented using either the built-in ODE integration tool `ode45` or `ode15s` in MATLAB™. The inner LPs were solved using CPLEX [16] and other LP solvers, such as the MATLAB™ `linprog` function and CLP in MATLAB's OPTI toolbox[6]. However, only results obtained using CPLEX are presented as this solver was the fastest in all test cases, although identical results, in terms of objective function value, were obtained in all of them.

The DAE systems resulting from the application of the Interior Point approach described in this work (Eqs. (16a) to (16f)) were solved using `ode15s` in MATLAB™. The Jacobians of the algebraic constraints were supplied as functions (analytically obtained) and the initial point of the DAE system was obtained using an implementation of the Primal-Dual Infeasible Interior-Point algorithm presented by Press et al. [28] to calculate the values of the fluxes, multipliers $\boldsymbol{\lambda}$, $\mathbf{y}$ and $\mathbf{z}$. The solution of the DAE system was also obtained using gPROMS (Process Systems Enterprise, Ltd., [29]). The system of implicit differential equations (18a) to (18c) was solved in gPROMS and MATLAB™ using `ode45`. The default value of the penalty parameter μ was set to 10^{-6} and `ode15s`, `ode45` and gPROMS were used with their default options.

The time required to perform the integration in MATLAB™ was obtained using the `timeit` function, that returns the wall-clock time, and also as the difference between two `cputime` calls, thus returning the total CPU time. In gPROMS the time required to calculate the trajectory was obtained, both as wall-time and CPU time, as the difference between a run where the integration time was set to reach the end of the simulation and a run where no integration was executed (zero integration time). This was necessary since gPROMS reports the execution time including the time required for system construction and analysis, solver loading and integration. In gPROMS the reported time values are an average of 10 runs.

Simulation problems were also solved using DFBAlab [11] using its default options for integration and LP optimization tolerances. CPLEX was used as the LP solver and the number of LP solved during a simulation was obtained as the number of calls to `cplex.solve` using the `profile` tool in MATLAB™.

All calculations were performed on a desktop computer equipped with an Intel® Core i7-6700 CPU and 32 GB RAM running Windows 10 64 bits.

4. Case studies

This section presents several case studies with increasing complexity. First, a fermentation where there is only one limiting substrate is analyzed for different metabolic networks. A second problem illustrates how our approach handles the changes in phenotypic phase planes [8] as the concentration of two substrates vary in a batch culture. As a third example, a medium scale metabolic model of *E. coli* is analyzed followed by a genome-scale one, as to asses the effect of the size of the metabolic network on the computational performance of the proposed approach. Finally, a fermentation where both the uptakes of oxygen and glucose are considered as inputs is analyzed using the genome-scale metabolic network of *S. cerevisiae* iND750 [7]. Finally, the application of the interior-point reformulation of dFBA problems to dynamical optimization problems is presented.

4.1. One limiting substrate, increasing metabolic network sizes.

4.1.1. Problem 1: Spirallus metabolic network

The metabolic network used in this example was taken from Zhao et al. [41] and includes only five intracellular metabolites (A to E) and three extracellular species (S, P and x). The purpose of this example is to serve as a motivating and instructive small case study which is easy to implement and reproduce by other researchers wishing to evaluate or adopt out proposed methodology in their own work. The graphic representation of the network can be found in Figure 2 in the work of Zhao et al. [41]. The time evolving

profile of the extracellular species concentration is given by the following differential equations:

$$\begin{aligned}
\frac{dx}{dt} &= v_5(t)x(t), \quad x(0) = 1.0 \text{ gL}^{-1}, \\
\frac{dS}{dt} &= -q_S(t)x(t), \quad S(0) = 20.0 \text{ mmolL}^{-1}, \\
\frac{dP}{dt} &= v_6(t)x(t), \quad P(0) = 0.0 \text{ mmolL}^{-1}, \\
q_S(t) &= q_{S,\max}\frac{S(t)}{S(t)+K_S},
\end{aligned} \tag{27}$$

with $K_s = 1.0$ mmolL^{-1} an affinity constant for the substrate and $q_{S,\max} = 3.8$ mmol(gDWh)$^{-1}$ its maximum uptake rate. The problem involves 7 fluxes and 6 metabolites, including the substrate uptake rate. The augmented stoichiometric matrix, $\mathbf{A}$, is given by:

$$A = \begin{array}{c} \\ A \\ B \\ C \\ D \\ E \\ S \end{array} \begin{array}{c} \begin{array}{ccccccc} v_1 & v_2 & v_3 & v_4 & v_5 & v_6 & q_S \end{array} \\ \begin{bmatrix} -1 & 0 & 0 & -1 & 0 & 0 & 1 \\ 1 & -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & -1 & 0 & 0 & 0 & 0 \\ -1 & 0 & 0 & 1 & -1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \end{array}$$

and

$$\mathbf{v} = (v_1, v_2, v_3, v_4, v_5, v_6, q_S)^T$$

is the vector of fluxes with lower and upper bounds given by

$$\mathbf{v}^{lo} = \left(0, 0, 0, -10^6, 0, 0, 0\right)^T \text{ mmol(gDWh)}^{-1},$$

$$\mathbf{v}^{up} = 10^6\,(1, 1, 1, 1, 1, 1, 1) \text{ mmol(gDWh)}^{-1}.$$

The problem is completed by defining:

$$\mathbf{b}(t) = \left(0, 0, 0, 0, 0, q_{S,\max}\frac{S(t)}{S(t)+K_S}\right)^T,$$

$$\frac{d\mathbf{b}(t)}{dt} = \left(0, 0, 0, 0, 0, \frac{q_{S,\max}}{(S(t)+K_s)^2}\frac{dS}{dt}\right)^T$$

and the objective weighting vector that ensures that biomass is maximized:

$$\mathbf{c} = (0, 0, 0, 0, 1, 0, 0)^T$$

The computational results are shown in Table 1. CPLEX and MATLAB's `linprog` LP solvers fail to produce a solution at 0.88 h of cultivation time when `ode45` is used. At this point, ODE45 calculates a negative value of substrate concentration causing an empty solution set for the inner LP. On the other hand, when using `ode15s`, no integration problems are encountered being CPLEX the LP solver requiring the shortest wall and CPU time to perform the simulation using 0.041 and 0.062 s, respectively. The use of our R-iODE approach allows for an even faster simulation, requiring only 0.016 s when `ode45` is used as the integrator. When gPROMS is used to perform the calculations, the overall wall time is 0.8 s, but this time includes the construction of the system of equations and its analysis while the time reported for `ode45` is only integration time. In this regard, gPROMS reports a nearly zero time to perform the integration, making this combination (R-iODE & gPROMS) the fastest for this problem.

The trajectories calculated using R-iODE, the Direct Approach and DFBAlab and the ones presented in Zhao et al. [41], are identical. Although the objective function used by Zhao et al. [41] is a non-linear one (the maximization of the biomass flux divided by the norm of the flux vector), the trajectories match. It is possible to show that, in this particular metabolic network, the linear objective function used in Problem 1

Table 1: Computational times and integration statistics for solving Problem 1 by the Direct or the Interior Point approaches and DFBAlab. The metabolic network in Problem 1 has 7 fluxes and 6 metabolites.

	Direct Approach		DFBAlab	Interior Point Approach			
Solver	CPLEX		CPLEX	DAE		R-iODE	
Integrator	ode45	ode15s	ode15s	ode15s	gPROMS	ode45	gPROMS
Wall time (s)	F[a][0.88h]	0.041	0.565	0.019	1(<1)[b]	0.016	1(<1)
CPU time (s)	-	0.062	0.531	0.047	0.040	0.046	0.032
N° of LPs solved	-	121	4	-	-	-	-
Infeasible LPs	-	0	0	-	-	-	-
Successful steps	-	50	105	87	59	25	60
Failed attemps	-	9	12	28	3	4	0
Function evals.	-	121	200	837	193	175	151
Jacobian evals.	-	3	1	19	16	0 (+18)[c]	12 (+18)

[a]Indicates that the method failed to complete the simulation at the time indicated in brackets

[b]Wall-clock time to construct and analyze the system, call solvers and integrate. In parenthesis, wall time for integration only.

[c]In parenthesis, the number of Jacobian evaluations required for solving the LP problem during initialization.

and the objective function used by Zhao et al. [41] are equivalent. It is interesting to note that the CPU and wall times required by DFBAlab to calculate the solution of problem 1 are longer than the ones required by the Direct Approach and the approaches developed in this work. This can be explained by the fact that DFBAlab requires checking at each time step whether a basic variable crosses zero or not, as described in Höffner et al. [15], then the basis set is not longer optimal and a new basis needs to be calculated by solving an LP problem. The event detection involves solving a pre-factorized linear systems of equations, and has an extra computational cost by comparison to R-iODE. Indeed, the CPU times reported by the MATLAB's Profiler tool are 0.274 seconds spent in `ode15s` execution and 0.233 seconds in the lexicographic optimization function of DFBAlab.

4.1.2. Problem 2: Metabolic network with changes in the phenotypic phase plane

A second illustrative example was constructed based on the metabolic network presented by Edwards et al. [8], which includes four phenotypes that can be reached depending on the uptake fluxes of oxygen and the carbon source (A). The metabolic network, extracellular mass balances and uptake rates required to represent the FBA problem as in Eq. 4 are given in Table 2 and Eq. 28. For irreversible reactions, the bounds on the fluxes are 0 and 100 mmol(gDWh)$^{-1}$, while for reversible reactions the bounds are -100 and 100 mmol(gDWh)$^{-1}$.

Table 2: Stoichiometry of the metabolic network in example 2, taken from Edwards et al. [8].

$\xrightarrow{q_A}$ A	ATP $\xrightarrow{R_{\text{ft}}}$
A + ATP $\xrightarrow{R_1}$ B	C + 10ATP $\xrightarrow{R_z}$ Biomass
B $\xrightarrow{R_2}$ 2ATP + 3NADH + C	$\xrightarrow{q_{O_2}}$ O_2
0.2C $\xrightarrow{R_3}$ 2NADH	C $\xleftrightarrow{C_{\text{out}}}$
C $\xleftrightarrow{R_4}$ ATP + 3D	D $\xleftrightarrow{D_{\text{out}}}$
C + 2NADH $\xrightarrow{R_5}$ 3E	E $\xleftrightarrow{E_{\text{out}}}$
NADH + O_2 $\xrightarrow{R_{\text{Res}}}$ 2ATP	

The uptake rate of the metabolite A is assumed to follow a Monod-type kinetics depending on the extracellular concentration of A (cA, gL^{-1}) while the uptake rate of oxygen was fixed at -15 mmol(gDWh)$^{-1}$, which in practice is equivalent to maintain a constant dissolved oxygen concentration.

$$
\begin{aligned}
\frac{dc_x}{dt} &= MW_z R_z(t) c_x(t), \quad c_x(0) = 1.0 \text{ gL}^{-1}, \\
\frac{dc_A}{dt} &= MW_A q_A(t) c_x(t), \quad c_A(0) = 10.0 \text{ gL}^{-1}, \\
\frac{dc_D}{dt} &= MW_D D_{out}(t) c_x(t), \quad c_D(0) = 0 \text{ gL}^{-1}, \\
\frac{dc_E}{dt} &= MW_E E_{out}(t) c_x(t), \quad c_E(0) = 0 \text{ gL}^{-1}, \\
q_{O_2}(t) &= -15 \text{ mmol(gDWh)}^{-1}, \\
q_A(t) &= -q_{A,\max} \frac{cA(t)}{cA(t) + K_A},
\end{aligned} \tag{28}
$$

with $K_A = 2.0$ gL^{-1} an affinity constant for the substrate and $q_{A,\max} = 12$ mmol(gDWh)$^{-1}$ its maximum uptake rate. Molecular weigths for biomass, substrate A, products D and E are given as $MW_z = 0.023$ g(mmol)$^{-1}$, $MW_A = 0.180$ g(mmol)$^{-1}$, $MW_D = 0.091$ g(mmol)$^{-1}$ and $MW_E = 0.046$ g(mmol)$^{-1}$. The values of the biomass flux (R_z) and product fluxes (D_{out} and E_{out}) are obtained as the solution of an LP problem where the linear constraints are given by the mass balances dictated by the stoichiometry presented in 2 and its bounds. The objective function of this problem is biomass maximization (flux R_z).

Table 3 shows the computational performance of the different methods tested for comparison and for the interior-point approaches. Every method tested detected the three changes in the phenotypic phase planes. Solution times were similar for the Direct and the R-iODE approaches, although the number of function evaluations differs. Although in the Direct Approach fewer function evaluations are required, they involve solving an LP problem at each call. On the other hand, for the R-iODE approach and the DAE approach, although more function evaluations are required, they are executed faster. As in Problem 1, DFBAlab requires more CPU and wall time to achieve a solution. Analysis of DFBAlab execution using the MATLAB's Profiler tool reveals that obtaining the basis of the LP problems and integration using ODE15s accounts for 75% of the CPU time. The solution of Problem 2 through the DAE formulation of the Interior Point Approach using `ode15s` required tightening the absolute integration tolerance from its default values to 10^{-8}. This change was necessary to produce a solution, but increased the solution times and the number of required steps.

Table 3: Computational performance obtained in the solution of Problem 2 by the Direct or the Interior Point approaches and DFBAlab. The metabolic network in Problem 2 has 13 fluxes and 10 metabolites.

	Direct Approach		DFBAlab	Interior Point Approach			
Solver	CPLEX		CPLEX	DAE		R-iODE	
Integrator	ode45	ode15s	ode15s	ode15s	gPROMS	ode45	gPROMS
Wall time (s)	0.062	0.064	0.478	0.201	1(<1)[a]	0.045	1(<1)
CPU time (s)	0.078	0.078	0.484	0.220	0.031	0.062	0.020
N° of LPs solved	211	169	4	-	-	-	-
Infeasible LPs	-	0	0	-	-	-	-
Successful steps	26	88	106	877	214	60	346
Failed attemps	9	16	1	287	2	17	5
Function evals.	211	169	203	2562	1012	463	1024
Jacobian evals.	-	2	4	159	167	118 (+9)[b]	12 (+9)

[a] Wall-clock time to construct and analyze the system, call solvers and integrate. In parenthesis, wall time for integration only.

[b] In parenthesis, the number of Jacobian evaluations required for solving the LP problem during initialization.

Figure 1.A presents the trajectories of the carbon source (A), biomass and products D and E obtained using the R-iODE approach in MATLAB. Figure 1.B shows the phenotypic phase plane of the biomass flux for the space of uptakes rates of A and oxygen in the region $[0, 10] \times [0, 20]$; mmol(gDWh)$^{-1}$.

The culture begins with a substrate concentration of 10 gL^{-1} producing an uptake rate of -10 mmol(gDWh)$^{-1}$, at a point located in the facet marked as P4 in the phenotypic phase plane. Region P4 is defined by an ex-

cess of carbon available compared to oxygen availability, thus the excess carbon flux was directed to the reduced product E. The culture stays in P4 for 3.65 h and then changes to a new region P3 characterized by the excretion of products D and E and the use of the cyclic reaction R3 to reduce the production of redox equivalents.

At 4.88 hours, the culture transitions from P3 to P2 as the uptake rate of A is further reduced. In this region, the product E is no longer produced as the production of D is sufficient to eliminate the redox equivalents under the prevailing oxygen uptake flux values. Finally, the phase plane P1 is a futile region where the electron acceptor (oxygen) is provided in excess, and the metabolic network dissipates the excess oxygen flux by using reaction R3, producing NADH, at the cost of oxidizing the precursor C. Since ATP is produced in excess, it is dissipated using reaction R_{ft}.

Figure 1.C shows the values of the multipliers enforcing the lower bounds of the fluxes, $\mathbf{y}$. The multipliers enforcing the upper bounds, $\mathbf{z}$, are not shown as they remain inactive during the simulation.

Elements of $\mathbf{y}$ that are active during the first 3.65 hours corresponds to the exchange fluxes of C and D, as these reactions remain inactive in the phase plane P4. A smooth transition to P3 is observed at this point, where the element of $\mathbf{y}$ for the exchange flux of D changes to a near-zero value as D starts being excreted. Conversely, the element of $\mathbf{y}$ for the ATP spillage reaction, R_{ft}, changes from zero to a positive value indicating an inactivation of this metabolic reaction.

At 4.88 hours, when the culture changes from the phenotypic phase plane P3 to P2, reactions R5 (producing E) and the excretion of E are inactivated, hence the elements of $\mathbf{y}$ corresponding to these reactions increase its value. Finally, at the transition from P2 to P1, D and E are inactivated and its elements in the vector $\mathbf{y}$ increase since the fluxes approximate their lower bounds.

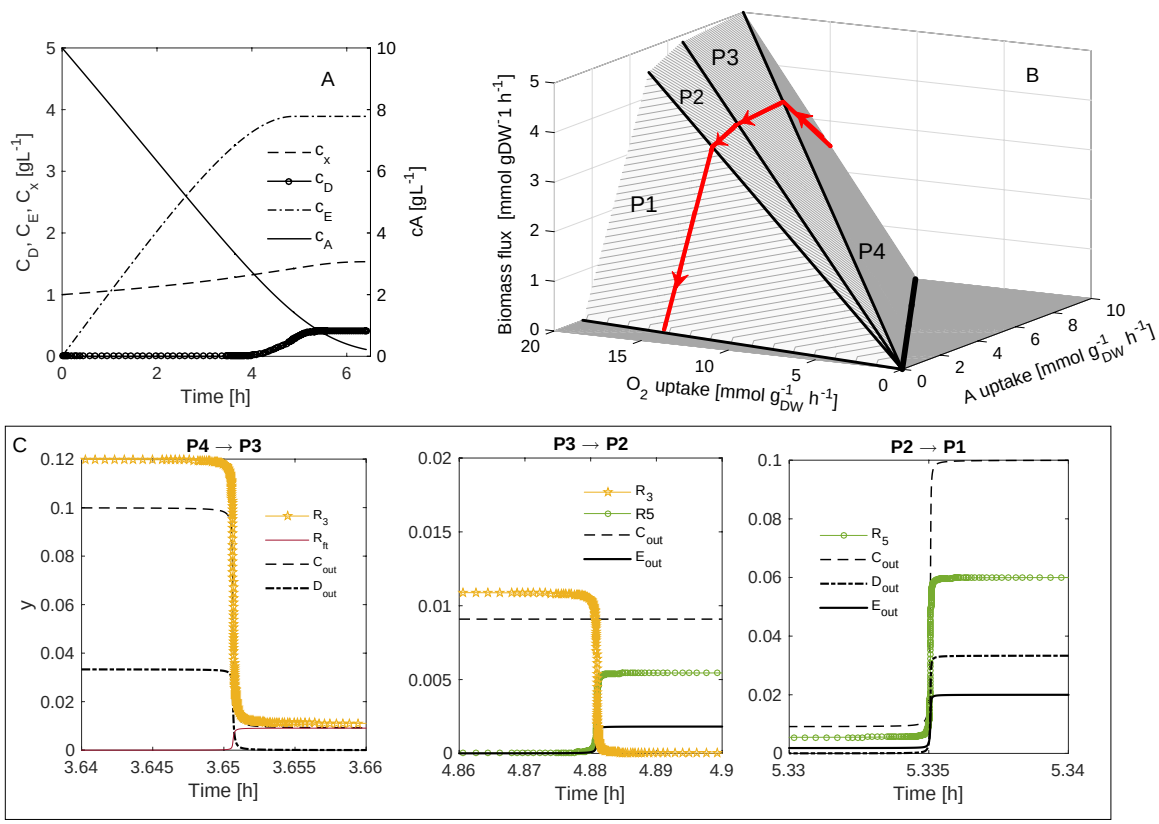

Figure 1: Trajectories for the differential variables in Problem 2 obtained by solving Eqs. (18a) to (18c) (R-iODE approach) using MATLAB™ ode45 with $\mu = 10^{-6}$ (panel A). Panel B shows the trajectory of the culture on the phenotypic phase plane. Panel C shows the changes in the values of selected elements of the vector enforcing the lower bounds, $\mathbf{y}$, as the culture changes from phenotypic phase plane.

4.1.3. Problem 3: E. coli *core model*

Problems 3 and 4 consider batch cultures where the time evolving profiles of the substrate (glucose) and product (ethanol) concentrations can be described by the following differential equations:

$$\begin{aligned}
\frac{dx}{dt} &= v_b(t)x(t), \quad x(0) = 1.0\,\mathrm{gL}^{-1}, \\
\frac{dS}{dt} &= \frac{180}{1000} q_S(t)x(t), \quad S(0) = 20.0\,\mathrm{gL}^{-1}, \\
\frac{dP}{dt} &= \frac{46}{1000} v_P(t)x(t), \quad P(0) = 0.0\,\mathrm{gL}^{-1}, \\
q_S(t) &= -q_{S,\max}\frac{S(t)}{S(t)+K_S},
\end{aligned} \tag{29}$$

where K_S is the affinity constant for glucose (1.0 gL^{-1}) and $q_{S,\max}$ the maximum specific uptake rate for glucose (10 $\mathrm{mmol(gDWh)}^{-1}$). In remaining problems in this work, the values of the specific growth rate, $v_b(t)$, and the specific products rates, $v_k(t)$ with k a selection of product fluxes such as ethanol or glycerol, are calculated as the solution of the following LP problem:

$$\begin{aligned}
&\min_{\mathbf{v}} -\mathbf{c}^T\mathbf{v} \\
\text{s.t.} \quad & \\
&\mathbf{A}\mathbf{v} = \mathbf{b}, \\
&\mathbf{v}^{up} - \mathbf{v} \geq 0, \\
&\mathbf{v} - \mathbf{v}^{lo} \geq 0, \\
&v_j = q_j(t),
\end{aligned} \tag{30}$$

where v_j is the uptake rate of j substrates in the metabolic model and is equal to the specific substrate consumption rate of the culture q_j. For example, in Problem 2, v_S, v_P and v_b are elements of vector $\mathbf{v}$.

The matrix $\mathbf{A}$ and the vectors $\mathbf{v}^{lo}$, $\mathbf{v}^{up}$, $\mathbf{c}$ and $\mathbf{b}$ are specific to each problem. In this problem (Problem 3), the stoichiometric matrix and bounds correspond to a flux balance representation of the central carbon metabolism of *E. coli* as published by [26], including 95 fluxes and 72 metabolites. The model is a subset of the genome-scale model *iAF1260* reported by Feist et al. [10]. The computational results for this problem, where biomass flux is maximized, are presented in Table 4. In terms of CPU time used for integration only, the R-iODE formulation running in gPROMS was the less demanding combination, followed by R-iODE with `ode45` and CPLEX with `ode45` in the Direct Approach. It is interesting to note that in Table 4 the number of function evaluations for the Direct Approach is smaller that the value reported for the Interior Point based methods. However, in the Direct Approach, each evaluation of the right hand side of the system of differential equations implies solving an LP. For the *E. coli* core model, each LP requires on average 60 iterations to reach a solution using the `linprog` algorithm in MATLAB™, while this number reduces to an average of 30 iterations when CPLEX is used. Similarly to the results found in Problems 1 and 2, DFBAlab shows higher CPU and wall time when compared to the Direct Approach and R-iODE.

The quality of the obtained solution is as relevant as the computational performance. Thereby, the trajectories of the biomass, substrate and product concentration calculated using DFBAlab and the R-iODE approach were compared. Figure 2.A shows the trajectories calculated using the R-iODE approach in gPROMS. Figure 2.B shows the average error between the glucose trajectories calculated using DBAlab and the R-iODE approaches while Figure 2.C shows the point-wise difference between the trajectories. The point-wise difference is defined, at a given time value t' in the integration time span as:

$$\Delta E_s(t') = S(t')_{\mathrm{DFBAlab}} - S(t')_{\mathrm{R-iODE}} \tag{31}$$

while the average error is given by:

$$\Delta \bar{E}_s[\%] = (t_f - t_0)^{-1} \int_{t_0}^{t_f} \Delta E_s(t)dt \tag{32}$$

As stated in Theorem 3.2, the gap between the trajectories vanishes as the value of μ approaches zero. This is also true for the duality gap (Figure 2.D).

Table 4: Comparison of the computational performance obtained by solving Problem 3 by the Direct or the Interior Point approaches and DFBAlab. The embedded FBA problem for this case study is composed of 95 fluxes and 72 metabolites and the biomass flux is maximized

	Direct Approach		DFBAlab	Interior Point Approach			
Solver	CPLEX		CPLEX	DAE		R-iODE	
Integrator	ode45	ode15s	ode15s	ode15s	gPROMS	ode45	gPROMS
Wall time (s)	0.047	0.038	0.339	0.054	1.1(<1)[a]	0.043	1.1(<1)
CPU time (s)	0.047	0.078	0.344	0.078	0.015	0.045	0.016
N° of LPs solved	68	48	4	-	-	-	-
Infeasible LPs	0	0	0	-	-	-	-
Successful steps	11	19	59	23	32	11	47
Failed attempts	0	0	11	15	1	0	0
Function evals.	68	49	111	85	92	67	113
Jacobian evals.	-	1	1	13	9	0 (+19)[b]	11 (+19)

[a]Wall-clock time to construct and analyze the system, call solvers and integrate. In parenthesis, wall time for integration only.

[b]In parenthesis, the number of Jacobian evaluations required for solving the LP problem during initialization.

4.1.4. Problem 4: Genome-scale metabolic model of E. coli *(iJR904)*

This example considers the same description of the fermentation kinetic as in Problem 4, in this way, the effect of a larger metabolic network over the computational performance of the Direct and Interior Point based approaches can be compared. The growth rate ($v_b(t)$) and the specific ethanol production rate ($v_P(t)$) at each time are obtained as the solution of the LP problem shown in Eq. (30). The stoichiometric matrix, objective function and bounds represents the genome-scale metabolic model of *E. coli* (iJR904 GSM/GPR) as reported by Reed et al. [32]. The model consists of 761 metabolites, 931 intracellular fluxes, 144 exchange fluxes and a flux representing biomass generation, which is maximized as the objective function of the embedded LP. Table 5 shows the computational performance of the Direct and Interior Point approaches. Once more, the R-iODE formulation implemented in gPROMS is the fastest one followed by the implementation of the R-iODE formulation using `ode45` and the Direct Approach using CPLEX, although the difference in CPU and wall time between the Direct Approach and DFBAlab is reduced when compared to Problem 1 and 3. We think this can be explained as follow. For embedded LP problems of small size (as in problems 1 and 3), obtaining its solution in CPLEX is faster than checking the optimality of the basis in DFBAlab at each integration time. Thus, for small problems, the Direct Approach should be faster than using DFBAlab, but, as the size of the LP problems increases, this difference reduces. For large scale problems, the performance of DFBAlab should be eventually better than the performance of the Direct Approach, since the computational cost of solving an LP is higher than the cost of solving a system of linear equations. Alternatively, using LP solvers such as CLP in MATLAB™ also reduces the difference in CPU and wall time between the Direct Approach and DFBAlab for problems 1, 3 and 4 (CLP wall times: 0.270, 0.146 and 1.488 seconds, respectively using ODE15s).

The DAE approach failed to produce a complete solution when using `ode15s` and gPROMS failed during initialization. The same initial point was used in both cases, which was calculated as the solution of an FBA problem at the starting time using an interior point method with the same barrier parameter value as the one used during dynamic simulation. From the theoretical point of view, the DAE formulation should have been solvable by any standard DAE solver using an exact Jacobian of the system both at the initialization and integration phases, as the Newton iteration involved is effectively the same as that within an interior-point solver would have used to solve the embedded optimization problem. The only significant difference that can be considered is that within an interior-point solver the line search step is safeguarded against violating the bounds (actually backtracking by a small scale factor from the step size that would render a bound exactly as active). This issue is not considered further in the current work and rather the full-ODE reformulation (R-iODE) also proposed in this work will be the main focus in the following case studies. Further investigation of the DAE formulation will be carried out in future work.

Figure 3 shows the trajectories of the differential variables obtained using gPROMS and its comparison with the trajectories calculated using the DFBAlab. As in Problem 3, the duality gaps (Figure 3.D) and the

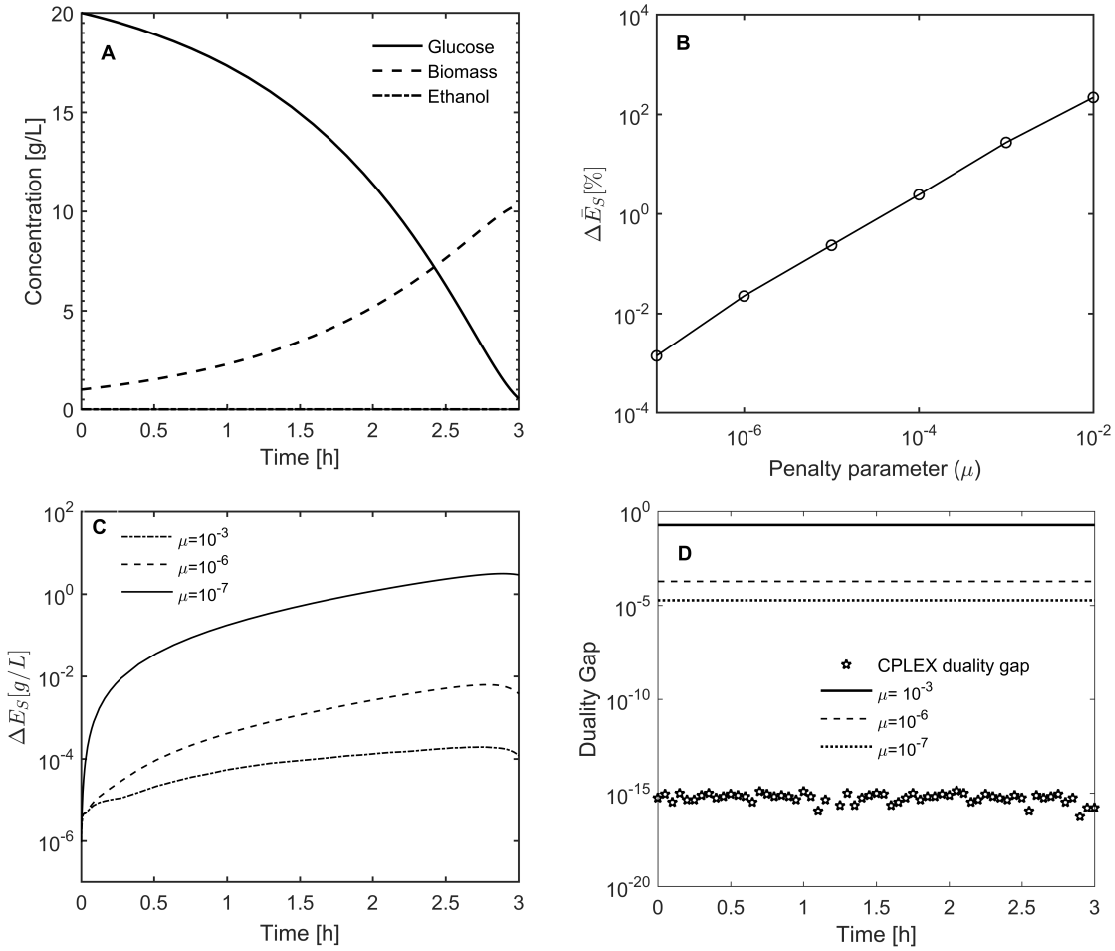

Figure 2: Trajectories for the differential variables in Problem 3 obtained by solving Eqs. (18a) to (18c) using gPROMS with $\mu = 10^{-6}$ (panel A). Panel B shows the average error between the glucose trajectories obtained using R-iODE and DFBAlab for different values of the barrier parameter μ, while panel C shows the pointwise difference between both methods for the glucose trajectories. Panel D shows the duality gap obtained for the direct solution approach using CPLEX (stars), and the ones obtained by solving Eqs. (18a) to (18c) with different values of the barrier parameter μ.

difference between the calculated trajectories at each time (Figure 3.C) decrease as the value of μ is reduced.

4.2. Problem 5: Ethanol production during a fed-batch culture of Saccharomyces cerevisiae

Industrial production of ethanol by fermentation is usually accomplished using the yeast *S. cerevisiae.* Hjersted et al. [14] used a kinetic model of the uptake of glucose and oxygen, coupled with the genome-scale metabolic model reported by Duarte et al. [7] to analyze the effects of an interruption of the supply of air in a fed-batch culture. The model considers that the specific uptake of glucose (G) and oxygen (O, as dissolved oxygen concentration) can be described as:

$$q_g = q_{g,max} \frac{G}{(G + K_g)} \frac{1}{(1 + E/K_{iE})}, \tag{33}$$

$$q_O = q_{O,max} \frac{O}{K_O + O}, \tag{34}$$

where K_O and K_g are saturation constants and K_{iE} is an inhibition constant for ethanol. The parameters $q_{g,max}$ and $q_{O,max}$ correspond to the maximum uptake rates of glucose and oxygen, respectively. Parameter

Table 5: Computational times and integration statistics for solving Problem 4 (761 metabolites, 1075 fluxes), where the biomass growth rate is maximized, by the Direct or the Interior Point approaches and DFBAlab.

	Direct Approach		DFBAlab	Interior Point Approach			
Solver	CPLEX		CPLEX	DAE		R-iODE	
Integrator	ode45	ode15s	ode15s	ode15s	gPROMS	ode45	gPROMS
Wall time (s)	F[2.9][a]	0.993	1.532	F[0.9]	F[0]	0.929	16(<1)[b]
CPU time (s)	-	1.000	1.593	-	-	0.930	0.234
N° of LPs solved	-	78	84	-	-	-	-
Infeasible LPs	-	0	0	-	-	-	-
Successful steps	-	28	113	-	-	11	70
Failed attemps	-	8	18	-	-	1	3
Function evals.	-	78	245	-	-	67	211
Jacobian evals.	-	3	5	-	-	0 (+18)[c]	21 (+18)

[a]Failed at the integration time indicated in brackets

[b]Wall-clock time to construct and analyze the system, call solvers and integrate. In parenthesis, wall time for integration only.

[c]In parenthesis, the number of Jacobian evaluations required for solving the LP problem during initialization.

values used during the simulation are shown in Table 6. The balances of the extracellular environment (the culture broth) are given by:

$$\frac{dV}{dt} = F, \quad V(0) = 0.5\,\mathrm{L} \tag{35}$$

$$\frac{d(xV)}{dt} = v_b(t)xV, \quad x(0) = 0.05\,\mathrm{gL}^{-1} \tag{36}$$

$$\frac{d(OV)}{dt} = k_L a(O_{sat} - O)V - q_O xV, \quad O(0) = 0.5O_{sat} \tag{37}$$

$$\frac{d(GV)}{dt} = FG_f - q_g xV, \quad G(0) = 10\,\mathrm{gL}^{-1} \tag{38}$$

$$\frac{d(EV)}{dt} = v_e xV, \quad E(0) = 0\,\mathrm{gL}^{-1} \tag{39}$$

where V is the liquid volume in the reactor at a given time, x is the biomass concentration, G_f is the glucose concentration in the feed (100 gL^{-1}) and F is the feed flow rate (0.044 Lh^{-1}). The simulation time is 16.0 h. Unlike the work presented by Hjersted et al. [14], a balance for the dissolved oxygen in the culture was included.

Table 6: Parameters used in Problem 5, taken from Hjersted et al. [14].

Variable	Value	Units
$q_{O,max}$	8	$\mathrm{mmol(gDWh)}^{-1}$
$q_{g,max}$	20	$\mathrm{mmol(gDWh)}^{-1}$
K_g	0.5	$\mathrm{g\,L}^{-1}$
K_{iE}	10	$\mathrm{g\,L}^{-1}$
K_O	0.003	$\mathrm{mmol\,L}^{-1}$
O_{sat}	0.30	$\mathrm{mmol\,L}^{-1}$

The specific growth rate and the specific ethanol production rate are calculated as the solution of an embedded FBA problem, where the flux of biomass is maximized, and its right hand side depends on the values of the substrates uptake rates $v_S(t)$ and $v_O(t)$. The metabolic model is comprised of 1059 metabolites and 1266 fluxes as described by Duarte et al. [7]. To allow for growth under anaerobic conditions, the lower bound of the following exchange reactions were set to -1000 $\mathrm{mmol(gDWh)}^{-1}$: `R_EX_ergst_e_`, `R_EX_zymst_e_`,`R_EX_hdcea_e_`, `R_EX_ocdca_e_`, `R_EX_ocdcea_e_` and `R_EX_ocdcya_e_`. Mimicking the

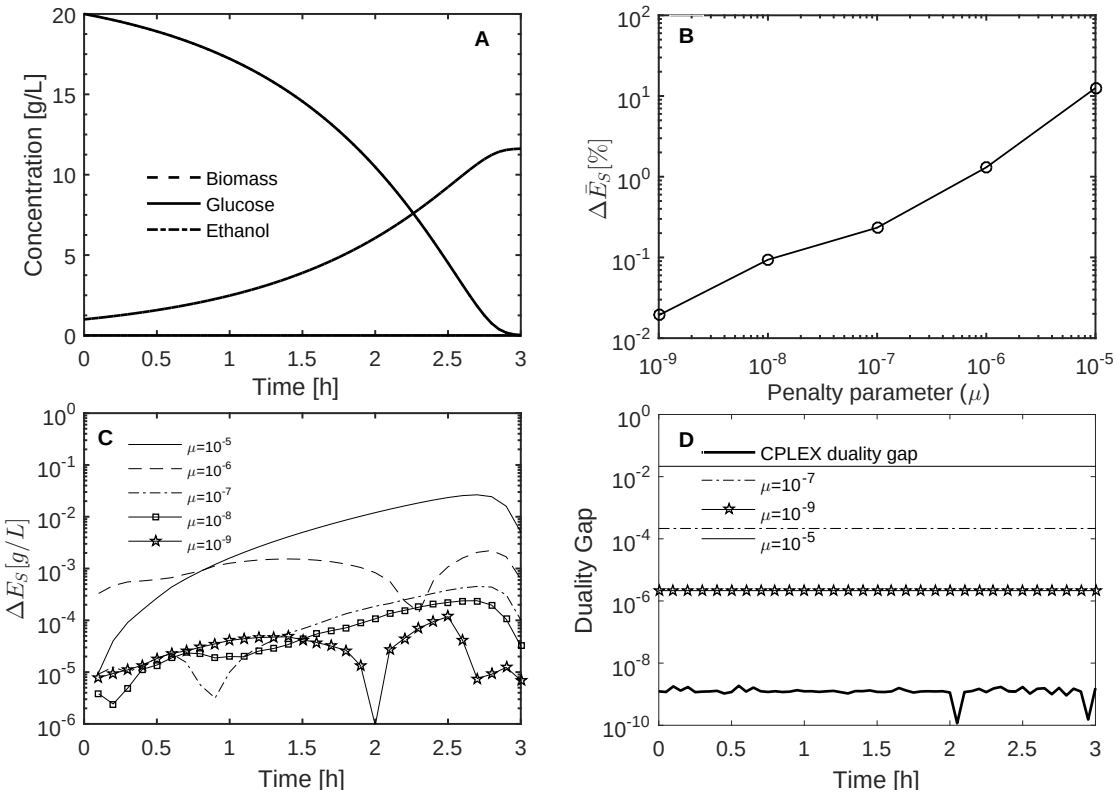

Figure 3: Trajectories for the differential variables in Problem 4 obtained by solving Eqs. (18a) to (18c) using gPROMS with $\mu = 10^{-6}$ (panel A). Panel B shows the average error between the glucose trajectories obtained using R-iODE and DFBAlab for different values of the barrier parameter μ, while panel C shows the pointwise difference between both methods for the glucose trajectories. Panel D shows the duality gap obtained for the direct solution approach using CPLEX (continuous line), and the ones obtained by solving Eqs. (18a) to (18c) with different values of the barrier parameter μ. When μ was reduced to 10^{-7} the integration tolerance was set to $10^{-}8$. In every other instance, gPROMS default tolerances were used.

simulations performed by Hjersted et al. [14], an step change at 7.7 h in the dissolved oxygen concentration from 50% saturation to anaerobic conditions was imposed. This change was forced by a changing the k_La value from 25 h^{-1} to zero. The trajectories of the differential variables for this simulation are shown in Figure 4.A for the trajectories obtained using DFBAlab and in panel B for the trajectories calculated using the R-iODE approach in gPROMS. They are not exactly the same as the ones presented in Hjersted et al. [14]. The difference can be explained by the fact that in our model the dissolve oxygen concentration reaches a near-zero value after five hours of culture, while in Hjersted et al. [14] step-change from 50% saturation of dissolved oxygen to anaerobic conditions at 7.7 h.

The computational results for the application of the Direct Approach, using CPLEX as the inner LP solver, DFBAlab and the Interior Point based approach are presented in Table 7. Results indicate that R-iODE & `ode45` was the most efficient approach to obtain the solution of the problem. Contrary to problems 1, 3 and 4, DFBAlab CPU and wall time for Problem 5 were approximately five times smaller than the ones obtained by using the Direct Approach. This can be explained by the use of the MATLAB's event detector function by DFBAlab, which allows stopping the integration at time 7.7 h (at the step-wise change in the oxygen concentration) and its reinitialization after the discontinuity. On the other hand, in our implementation of the Direct Approach, no event detection was used and the integration was forced to continue during the step-wise change in the oxygen concentration. This results in 484 LP problems solved during the integration in the Direct Approach using ODE15s compared to 22 LPs solved by DFBAlab.

Table 7: Solution summary for Problem 5 including results obtained using the Interior Point or the Direct approaches and DFBAlab. The embedded LP problem has 1059 metabolites and 1266 fluxes, biomass flux is maximized.

	Direct Approach		DFBAlab	Interior Point Approach	
Solver	CPLEX		CPLEX	R-iODE	
Integrator	ode45	ode15s	ode15s	ode45	gPROMS
Wall time (s)	11.32	7.64	1.68	1.714	27.8(<1)[a]
CPU time (s)	10.81	6.80	2.01	5.129	0.372
N° of LPs solved	724	484	22	-	-
Infeasible LPs	0	0	0	-	-
Successful steps	110	149	226	26	137
Failed attemps	26	39	12	0	21
Function evals.	724	484	606	136	422 (+70)
Jacobian evals.	-	20	23	0 (+70) [b]	45

[a]Wall-clock time to construct and analyze the system, call solvers and integrate. In parenthesis, wall time for integration only.

[b]In parenthesis, the number of Jacobian evaluations required for solving the LP problem during initialization.

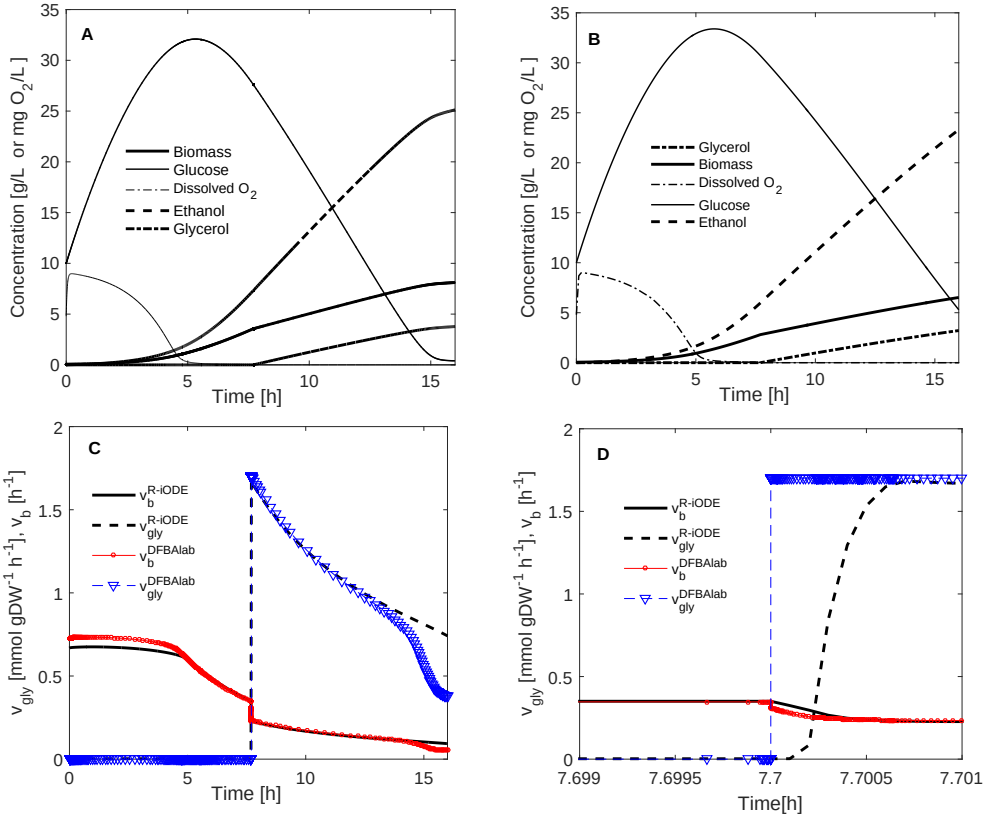

Figure 4: Biomass, glucose and ethanol time profiles for Problem 5. A stepwise change in the dissolved oxygen concentration was forced at 7.7 h by imposing a change in the value of k_La. The simulation in panel A was obtained using DFBAlab (maximizing biomass, ethanol and glycerol production using lexicographic optimization) while the simulation in panel B was obtained using the R-iODE approach implemented in gPROMS. Panel C shows the values of biomass specific growth rate and glycerol flux calculated using the R-iODE approach and DFBAlab for the complete integration time span, while panel D shows those fluxes around 7.7 hours.

Figure 4.C and Figure 4.D shows the fluxes of biomass and glycerol during the simulation. At 7.7 hours, when the k_La value is changed to zero, the basis of the inner LP problem changes. Consequently the values of the biomass and glycerol fluxes change abruptly but continuously. However, Figure 4.D shows that this

change is, in fact, continuous. On the other hand, DFBAlab [11] formulation relies on the fact that within a phenotypic phase plane [8], the objective changes proportionally to changes in the uptake rates. This means that within the phenotypic phase plane, the solution basis does not change. During the dynamics of a fermentation, this allows recalculation of the fluxes' values without the need of further LP solution, rather a simple back-substitution provides their values. When the culture moves to a new phenotypic phase plane, a new solution basis must be determined and factorized to be reused accordingly until the next potential basis change. As shown in Figure 4.D, the change from one basis to another is discontinuous when DFBAlab is used.

The active set selection, or equivalently, the basis selection, has nothing to do with the use of an interior point method, as the active vertex solution of an LP is approached from the interior of the feasible region of the inequality constraints. As such, our approach is a smoothing of the LP problem as is standard in all interior point methods. There is no active set, or basis, to deal with in the proposed methodology in our work. The interior point formulation, embedded within a dynamic system, will exhibit no discontinuities in base changes, which would be reflected by abrupt changes in the Lagrange multiplier values. Instead, the interior-point formulation smooths out the trajectories of moving from the vicinity of one active vertex to the next active vertex when this happens, which corresponds to the change of basis of the original dFBA formulation as implemented in DFBAlab.

It is noted that in the simulations reported by Harwood et al. [12] and Hjersted et al. [14] using lexicographic optimization and the Direct Approach, respectively, glycerol is produced during the anaerobic phase of the culture. In fact, the flux of glycerol is not uniquely determined and the glycerol concentration can take any value between zero and the maximum glycerol trajectory shown in 4.A. Vargas et al. [36], obtained glycerol production by modifying the ATP maintenance flux through a coupling of this variable with the nitrogen uptake rate.

In our current implementation, the solution of the barrier-formulated LP problem will tend, from the interior of the feasible region, to the analytic center of the facet defined by the alternative vertices giving the same solution, in the case the objective function hyperplane is parallel to a facet of the feasible set polytope [19]. The analytic center of a facet of the feasible polytope of the embedded FBA problem, is the barrier problem without the objective function and taking into account only the active set of bounds defining the 'active' facet subject to the equality constraints (flux balances). As the analytic center of each facet of the feasible set polytope is a uniquely defined point, the dFBA approach as reformulated via an interior-point method will always give a unique solution to the underlying embedded optimization problems at each time of the simulation horizon. It is noted that this solution is one of many possible solutions, and the fact that it is unique is artificial to the specific methodology – yet very convenient for the purpose of simulation This artificiality is not unique to our methodology. In fact, in the methodology adopted by DFBAlab [11], the nonuniqueness of a solution is dealt with a lexicographic optimization. This is based on an arbitrary choice of substrates which are prioritized accordingly to previous knowledge of substrate uptakes and products output involved in the metabolic process.

4.3. Dynamic optimization problems

The formulation proposed in this work for the solution of dFBA problems, as for example given in Eqs. (18a) to (18c), results in a smooth ODE simulation problem. As such, the simulation problem can be embedded within any continuous optimization algorithm to provide function and gradient evaluations towards calculation of optimal profiles for control variables for time-evolving biochemical processes. This is demonstrated with two illustrative case studies in this section.

It is noted that the methodology proposed by Höffner et al. [15] and Harwood et al. [12], and implemented in the computational package DFBAlab [11], as reviewed in the Introduction section of this paper, leads to the most direct way to solve dFBA simulation problems. However, as it was explained in the same section, the methodology relies on the identification of LP base changes during the integration process − which may occur correctly when LPs are embedded in a dynamic system. This identification of base changes constitutes implicit discontinuity event detections, which although leads to C^0 continuous state variable profiles, these are nonetheless non-differentiable at the points in time where the events take place (i.e., where the LP base changes).

Solution of optimal control problems requires the underlying dynamic models to not contain implicit discontinuities in order for the associated function evaluations to be differentiable with respect to the opti-

mization parameters of the models (or those parameterizing control functions). As such, to our knowledge there has been no other smooth simulation model approach proposed in the open literature to this date other than the methodology being put forward in this work that has this required property −as a result of the interior point method transformation used, and although the smoothing introduced results in an approximate solution of the dFBA problems. This allows the efficient and robust optimization of dFBA models through the feasible path approach for optimal control problems [38, 39]. The only alternative way to ensure a differentiable approach that can guarantee the smooth solution of associated dFBA optimal control problems is via collocation [13], as reviewed in the Introduction section, which however results in requirement to solve optimization problems of prohibitive size for any realistic metabolic network model.

4.3.1. Problem 6: E. coli *fed-batch fermentation considering substrate inhibition*

This example corresponds to a dynamic optimization problem where a piecewise constant feed flow rate profile is optimized to maximize the mass of *E. coli* cells at the end fermentation time in a fed-batch culture:

$$
\begin{aligned}
&\max_{F(\cdot)} (xV)|_{t_f} \\
&\text{s.t.} \qquad \frac{dV}{dt} = F, \quad V(0) = 1.0\,L \\
&\qquad\qquad \frac{dx}{dt} = v_b x - x\frac{F}{V}, \quad x(0) = 1.0\,\text{gL}^{-1} \\
&\qquad\qquad \frac{dS}{dt} = (S_F - S)\frac{F}{V} + \frac{180}{1000} q_s x, \quad S(0) = 2.0\,\text{gL}^{-1} \\
&\qquad\qquad \frac{dP}{dt} = \frac{46}{1000} v_P X - P\frac{F}{V}, \quad P(0) = 0.0\,\text{gL}^{-1} \\
&\qquad\qquad q_s = -q_{s,max}\frac{S}{S + K_S + S^2/K_I}
\end{aligned}
\tag{40}
$$

with $q_{s,max} = 10$ mmol(gDWh)$^{-1}$, $K_S = 1.0$ gL^{-1} and $K_I = 10.0$ gL^{-1}. Glucose is fed to the reactor at a concentration of $S_F = 100$ gL^{-1} at a piecewise constant rate $F(t)$ that is determined by solving the optimal control problem stated in Eq. (40). The specific growth rate of *E. coli* and the ethanol specific production rate (v_P) are determined from the metabolic network model of the *E. coli* central metabolism [26], previously described in Problem 3. The flux balance model is an LP problem of the form presented in Eq. (30) and biomass flux is maximized. Thus, the solution of the LP problem is connected to the solution of the optimal control problem by the exchange flux of glucose v_S. The optimal control problem was solved using two approaches: (i) by a Direct Approach where the inner LP problem is solved using CPLEX during the integration by `ode45` in MATLAB™ and the optimal profile is determined by MATLAB's `fmincon`, and (ii) by appending the ODEs derived by applying the R-iODE approach to an optimal control problem in gPROMS. While in (ii), gPROMS can calculate the gradient of the objective function with respect to the piecewise constant values of the flow rate during the integration using sensitivity equations, in (i) `fmincon` estimates by numerical differentiation. Thus, one would expect an improved performance of gPROMS over `fmincon`. Table 8 shows the time required to achieve a solution and the objective function value. Results indicate that, as expected, gPROMS requires less CPU time compared to MATLAB's `fmincon`, and produces also a significantly better optimal solution in terms of the objective function value.

Table 8: CPU time, NLP solver major iterations and the value of the objective function obtained for the solution of Problem 6 (95 fluxes and 72 metabolites) for five and ten control intervals. The objective function of the embedded LP problem is the maximization of the biomass flux and the objective function of the dynamic optimization problem is the mass of *E. coli* cells at the end time.

	`ode45` & CPLEX & `fmincon`		R-iODE & gPROMS	
Number of control intervals	5	10	5	10
CPU time (s)	266.4	1027.6	123.8	107.6
NLP iterations	15	25	52	55
Objective function	485.4	473.7	596.7	669.3

The trajectory of the objective function and the calculated feed flow rate profiles are shown in Figure 5. The profile of the optimal feed flow rate can be analyzed by considering that the solution of the problem

is to maintain the glucose specific uptake rate at its maximum. Using Eq. (40) which defines q_s it can be calculated that the maximum glucose uptake is achieved at a concentration of 3.2 gL^{-1} of glucose. Thereby, an optimal feed flow rate will maintain the glucose concentration as close to this value as possible, which necessarily implies that the feed flow profile will be exponential as a consequence of the exponential growth rate of the culture. As shown in Figure 5.B, the profile calculated by gPROMS approaches an exponential one and allows for an average substrate concentration during the culture of 3.8 gL^{-1}.

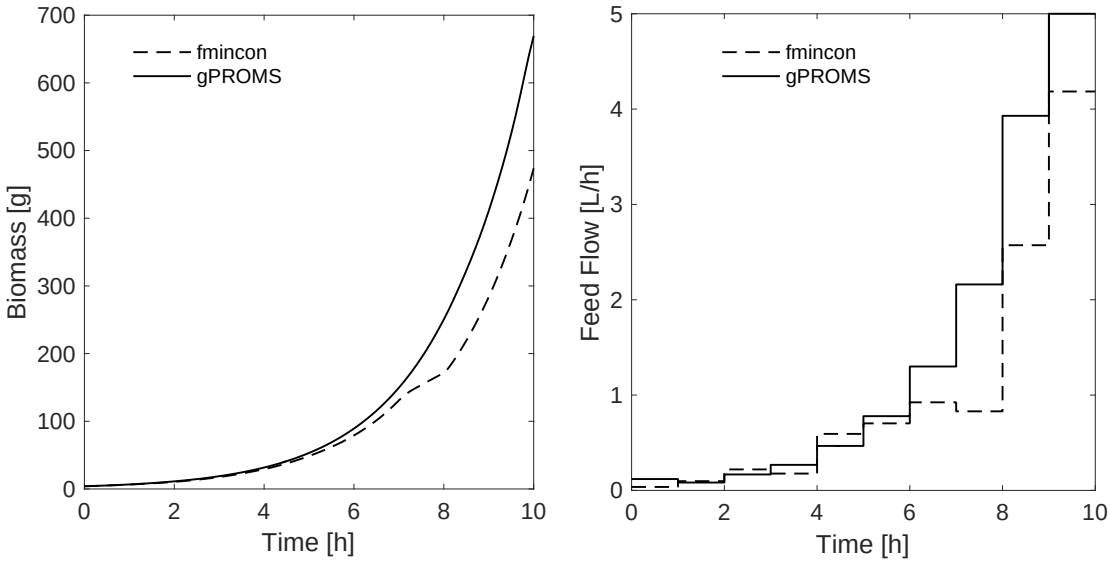

Figure 5: Trajectories for the accumulated biomass (A) and the calculated optimal feed flow rate profile (B) in Problem 6. for 10 piecewise constant control elements obtained using gPROMS and MATLAB's `fmincon`

4.3.2. Problem 7: Maximization of ethanol production in a fed-batch culture of S. stipitis

During the last decade, ethanol production from sugars obtained from lignocellulose, a natural polymer composed of cellulose, lignin and hemicellulose has been the subject of intensive investigation [21]. While glucose fermentation by *S. cerevisiae* is a mature technology, the fermentation of xylose, one of the sugars released by the depolymerization of hemicellulose in certain species of plants, is still in development. In this regard, Slininger et al. [35] proposed a dynamic model describing the growth and ethanol production by *Scheffersomyces* (*Pichia*) *stipitis* fed with xylose as the sole carbon and energy source. The unstructured kinetic model describes the growth of the yeast (viable and total cells) and the dynamics of xylose and oxygen consumption as well as the production of ethanol as a system of linear differential equations. The model, coupled with an FBA description of the metabolism, was used for the formulation of an optimal control problem whose objective function is the maximization of the ethanol mass produced at the end time of the fermentation and in a reactor with a maximum operation volume of 8.0 L.

$$
\begin{aligned}
&\max_{F(\cdot),k_La} (PV)|_{t_f} \\
&\text{s.t.} \qquad \frac{dV}{dt} = F, \quad V(0) = 1.0\,L \\
&\qquad \frac{dx_T}{dt} = v_b x_T - x_T\frac{F}{V}, \quad x_T(0) = 1.0\,\mathrm{gL}^{-1} \\
&\qquad \frac{dx}{dt} = v_b(1-f_d)x - x\frac{F}{V}, \quad x(0) = 1.0\,\mathrm{gL}^{-1} \\
&\qquad \frac{dS}{dt} = (S_F - S)\frac{F}{V} + \frac{150}{1000}q_s x, \quad S(0) = 2.0\,\mathrm{gL}^{-1} \\
&\qquad \frac{dP}{dt} = \frac{46}{1000}v_P x - P\frac{F}{V}, \quad P(0) = 0.0\,\mathrm{gL}^{-1} \\
&\qquad \frac{dO_2}{dt} = k_La(C_{sat} - O_2) - 32q_{O_2}x - O_2\frac{F}{V}, \quad O_2(0) = 0.0\,,\mathrm{gL}^{-1} \\
&\qquad \mu_b = \mu_m\left(\frac{S}{K_s+S} - \frac{K_i}{K_i+S_m-S}\right)\left(1-\left(\frac{P}{P_m}\right)^A\right)\left(\frac{O_2}{K_{ox}+O_2}\right), \\
&\qquad f_d = (0.194 + 0.000381\cdot S)(1 - 0.00356\cdot P + 0.000555\cdot P^2), \\
&\qquad C_{sat} = 0.21\cdot 1.08\cdot(34.6 - 0.0644\cdot S + 0.000156\cdot S^2), \\
&\qquad \beta = \beta_m\cdot\left(e^{-S/K_{iP}} - e^{-S/K_{SP}}\right)\left(1-\left(\frac{P}{P_{mP}}\right)^B\right), \\
&\qquad q_s = \frac{\alpha\mu_b + \beta}{0.421 - 0.343\mu_b}, \\
&\qquad q_{O_2} = \frac{\mu_b}{Y_{O_2}}.
\end{aligned}
\tag{41}
$$

Model parameters are given in Table 9 and were taken from Slininger et al. [35]. The xylose concentration in the feed flow was set at 50 gL^{-1}. In the original model of Slininger et al. [35], the ethanol specific production rate and the specific growth rate and are given as:

$$
\begin{aligned}
v_P = \frac{1000}{46}q_P = \frac{1000}{46}(\alpha\mu_b + \beta), \\
v_b = \mu_b.
\end{aligned}
$$

On the other hand, we coupled the unstructured kinetic model from Slininger et al. [35] with the metabolic network model presented by Balagurunathan et al. [1], defined by matrix $\mathbf{A}$ and vectors $\mathbf{b}$, $\mathbf{v}^{up}$ and $\mathbf{v}^{lo}$. Thus, the specific growth rate (v_b) and the specific ethanol production rate (v_P) are determined by solving an embedded LP defined by Eq. (30).

Optimization runs were done for a fermentation time span of 15 hours and considering 20 control intervals and two control variables, namely, the feed flow rate and the oxygen feed flow rate (this was done indirectly by taking the k_La as control). This lead to an optimal ethanol production of 139.5 g. Since this value is only marginally higher than the one obtained by only optimizing the feed flow rate as a piecewise constant function and the k_La as a fixed value (see Table 10) only the latter results will be presented.

Table 9: Parameters used in Problem 7, taken from Slininger et al. [35].

Variable	Value	Units
A	1.32	–
B	0.935	–
P_m	64.3	$\mathrm{g\,L^{-1}}$
P_{mP}	189	$\mathrm{g\,L^{-1}}$
K_i	60.2	$\mathrm{g\,L^{-1}}$
K_{iP}	72.7	$\mathrm{g\,L^{-1}}$
K_{ox}	0.1	$\mathrm{mg\,L^{-1}}$
K_S	0.36	$\mathrm{g\,L^{-1}}$
K_{SP}	45.91	$\mathrm{g\,L^{-1}}$
S_m	253	$\mathrm{g\,L^{-1}}$
Y_{O_2}	0.00270	$\mathrm{g\,mg^{-1}}$
β_m	1.43	$\mathrm{g\,gh^{-1}}$
α	1.43	$\mathrm{g\,g^{-1}}$
μ_{max}	0.71	$\mathrm{h^{-1}}$

Table 10: Solution summary for Problem 7 including CPU time, NLP iterations and the value of the objective function. The embedded FBA problem has 1371 fluxes and 971 metabolites. The objective function of the embedded FBA problem is biomass flux, while the objective function maximized in the dynamic optimization problem is the mass of ethanol at the end time.

Controls	F	Piecewise	Piecewise
	k_La	Time-invariant	Piecewise
CPU time (s)		223.0	863.5
NLP iterations		29	37
Ethanol (PV (g))		137.7	139.5

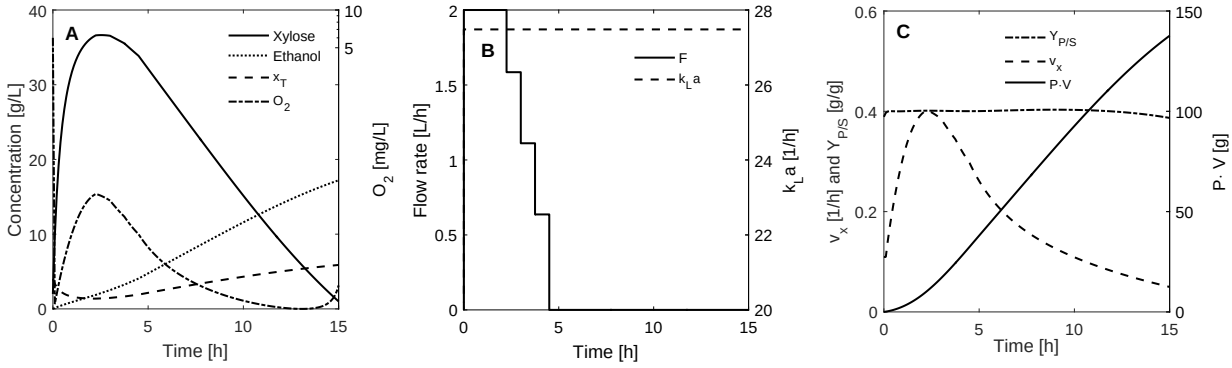

Figure 6: Trajectories for the state variables and controls in Problem 7. Panel A shows the trajectories of ethanol, xylose, biomass and oxygen. Panel B presents the profile of the control variables, while panel C shows the values of the specific growth rate, ethanol yield and the objective function.

Figure 6.B presents the optimal profiles for the xylose fed and the optimal k_La value for fed batch-culture. Panel A, shows the xylose, ethanol, total biomass and oxygen concentration profiles calculated, and panels C displays the trajectories for the ethanol yield, specific growth rate and the accumulated mass of ethanol. To the best of our knowledge, Problem 7 represents the largest ever dynamic optimization problem with an embedded dFBA model solved with 1371 fluxes and 971 metabolites. Previously, Hjersted and Henson [13] optimized the feed flow rate profile of a fed-batch culture of *S. cerevisiae* using the DOA approach, with the dFBA model used being a representation of the central carbon metabolism with 82 fluxes.

5. Conclusions

This work presents a new approach for solving dynamic flux balance analysis problems. The approach replaces the embedded linear programming problem by the first order optimality conditions of an equivalent problem where the bounds on the fluxes values are handled by logarithmic barrier functions. Based on theoretical results from the duality theory of linear programming, we show that the system of differential equations with an embedded LP, the typical formulation of a dFBA problem, can be converted into a system of implicit ordinary differential equations that can be solved efficiently using standard integration methods.

The proposed approach was shown to produce a uniquely determined trajectory that can be made arbitrarily close to the exact trajectory of the dFBA problem by reducing the size of a penalty parameter. The proposed approach was tested by applying our interior point based formulation for the solution of dynamic flux balance analysis on six examples obtained from the open literature, and the results show that the method presented in this work is highly competitive, in terms of computational time, all other methodologies and solvers tested, even when highly efficient LP solvers such as CPLEX are used. Moreover, the method was used to solve to the best of our knowledge, for the first time, a dynamic optimization problem (optimal control problem) with a genome-scale dFBA model embedded using the advanced process simulation package gPROMS.

Future work will continue exploring the capabilities offered by the new methodology, such as applications to real-world large-scale metabolic networks, co-cultures including several species of microorganisms, parameter estimation in kinetic models embedding FBA to enhance their predictive ability, and further investigation and generalization of including nonlinear convex objective functions in dFBA. Also, the issue of the numerical challenges posed by the DAE formulation of dFBA proposed in this work, as discussed in Section 4.1.4, will be also further investigated. Finally, our approach to solve dFBA problems by inclusion into standard process simulation packages opens up the scope for further applications to include dFBA within entire plant simulation models.

6. Acknowledgments

- R. Conejeros would like to thank CONICYT's research grant FONDECYT 1151295 for funding this research.
- F. Scott gratefully acknowledges financial support from CONICYT (Proyectos REDES ETAPA INICIAL, Convocatoria 2017, REDI170254).

References

[1] Balagurunathan, B., Jonnalagadda, S., Tan, L., Srinivasan, R., 2012. Reconstruction and analysis of a genome-scale metabolic model for *Scheffersomyces stipitis*. Microbial Cell Factories 11 (27), 1–18.

[2] Baumrucker, B., Renfro, J., Biegler, L., dec 2008. MPEC problem formulations and solution strategies with chemical engineering applications. Computers & Chemical Engineering 32 (12), 2903–2913.

[3] Bazaraa, M., Hanif, H., Shetty, C., 2006. Nonlinear programming: theory and algorithms, 3rd Edition. John Wiley & Sons.

[4] Biegler, L. T., nov 2007. An overview of simultaneous strategies for dynamic optimization. Chemical Engineering and Processing: Process Intensification 46 (11), 1043–1053.

[5] Chung, B. K., Selvarasu, S., Andrea, C., Ryu, J., Lee, H., Ahn, J., Lee, H., Lee, D.-y., 2010. Genome-scale metabolic reconstruction and in silico analysis of methylotrophic yeast Pichia pastoris for strain improvement. Microbial Cell Factories 9 (1), 50.

[6] Currie, J., Wilson, D. I., 2012. OPTI: Lowering the barrier between open source optimizers and the industrial MATLAB user. In: Foundations of Computer-Aided Process Operations. Savannah, Georgia, USA.

[7] Duarte, N. C., Herrgård, M. J., Palsson, B. Ø., 2004. Reconstruction and validation of *Saccharomyces cerevisiae* iND750, a fully compartmentalized genome scale metabolic model. Genome Research 14 (7), 1298–1309.

[8] Edwards, J., Ramakrishna, R., Palsson, B., 2002. Characterizing the metabolic phenotype: A phenotype phase plane analysis. Biotechnology and Bioengineering 77 (1), 27–36.

[9] Emenike, V. N., Schulze, M., Schenkendorf, R., Krewer, U., 2017. Model-based optimization of the recombinant protein production in Pichia pastoris based on dynamic flux balance analysis and elementary process functions. In: 27th European Symposium on Computer Aided Process Engineering. Vol. 40. Elsevier Masson SAS, pp. 2815–2820.

[10] Feist, A. M., Henry, C. S., Reed, J. L., Krummenacker, M., Joyce, A. R., Karp, P. D., Broadbelt, L. J., Hatzimanikatis, V., Palsson, B. Ø., 2007. A genome-scale metabolic reconstruction for *Escherichia coli* K-12 MG1655 that accounts for 1260 ORFs and thermodynamic information. Molecular Systems Biology 3.

[11] Gomez, J. A., Höffner, K., Barton, P. I., 2014. DFBAlab: a fast and reliable MATLAB code for dynamic flux balance analysis. BMC Bioinformatics 15 (1), 409.

[12] Harwood, S. M., Höffner, K., Barton, P. I., 2016. Efficient solution of ordinary differential equations with a parametric lexicographic linear program embedded. Numerische Mathematik 133 (4), 623–653.

[13] Hjersted, J. L., Henson, M. A., 2006. Optimization of fed-batch *Saccharomyces cerevisiae* fermentation using dynamic flux balance models. Biotechnology Progress 22 (5), 1239–1248.

[14] Hjersted, J. L., Henson, M. A., Mahadevan, R., 2007. Genome-scale analysis of *Saccharomyces cerevisiae* metabolism and ethanol production in fed-batch culture. Biotechnology and Bioengineering 97 (5), 1190–1204.

[15] Höffner, K., Harwood, S. M., Barton, P. I., 2013. A reliable simulator for dynamic flux balance analysis. Biotechnology and Bioengineering 110 (3), 792–802.

[16] IBM ILOG CPLEX, 2013. CPLEX Optimization Studio, Users Manual. IBM, Armonk, NY.

[17] Kaplan, U., Türkay, M., Biegler, L., Karasözen, B., may 2009. Modeling and simulation of metabolic networks for estimation of biomass accumulation parameters. Discrete Applied Mathematics 157 (10), 2483–2493.

[18] King, Z. A., Lloyd, C. J., Feist, A. M., Palsson, B. O., 2015. Next-generation genome-scale models for metabolic engineering. Current Opinion in Biotechnology 35, 23–29.

[19] Kojima, M., Mizuno, S., Yoshise, A., 1989. Progress in Mathematical Programming, Interior-Point and Related Methods. Springer-Verlag, Ch. A primal-dual interior point method for linear programming, pp. 29–47.

[20] Lu, T. T., Shiou, S. H., 2002. Inverses of 2 X 2 block matrices. Computers and Mathematics with Applications 43 (1-2), 119–129.

[21] Lynd, L. R., Liang, X., Biddy, M. J., Allee, A., Cai, H., Foust, T., Himmel, M. E., Laser, M. S., Wang, M., Wyman, C. E., 2017. Cellulosic ethanol: status and innovation. Current Opinion in Biotechnology 45, 202–211.

[22] Mahadevan, R., Edwards, J. S., Doyle, F. J., 2002. Dynamic flux balance analysis of diauxic growth in *Escherichia coli*. Biophysical Journal 83 (3), 1331–1340.

[23] Megiddo, N., 1989. Pathways to the optimal set in linear programming. Springer New York, New York, NY.

[24] Monteiro, R. D. C., Adler, I., 1989. Interior path following primal-dual algorithms. part I: Linear programming. Mathematical Programming 44 (1-3), 27–41.

[25] Nocedal, J., Wright, Stephen, J., 2006. Numerical optimization. Springer-Verlag, USA.

[26] Orth, J. D., Palsson, B. Ø., Fleming, R. M. T., 2010. Reconstruction and use of microbial metabolic networks: the core *Escherichia coli* metabolic model as an educational guide. EcoSal Plus 4 (1).

[27] Orth, J. D., Thiele, I., Palsson, B. Ø., 2010. What is flux balance analysis? Nature Biotechnology 28 (3), 245–248.

[28] Press, W. H., Teukolsky, S. A., Vetterling, W. T., Flannery, B. P., 2007. Numerical recipes 3rd edition: The art of scientific computing. Vol. 3. Cambridge University Press, Cambridge.

[29] Process Systems Enterprise Limited, 2016. gPROMS ModelBuilder Documentation.

[30] Raghunathan, A. U., Biegler, L. T., jan 2005. An Interior Point Method for Mathematical Programs with Complementarity Constraints (MPCCs). SIAM Journal on Optimization 15 (3), 720–750.

[31] Raghunathan, A. U., Pérez-Correa, J. R., Agosin, E., Biegler, L. T., nov 2006. Parameter estimation in metabolic flux balance models for batch fermentation- Formulation & Solution using Differential Variational Inequalities (DVIs). Annals of Operations Research 148 (1), 251–270.

[32] Reed, J. L., Vo, T. D., Schilling, C. H., Palsson, B. O., 2003. An expanded genome-scale model of *Escherichia coli* K-12 (iJR904 GSM/GPR). Genome Biology 4 (9), 54.

[33] Savinell, J. M., Palsson, B. O., 1992. Network analysis of intermediary metabolism using linear optimization. I. Development of mathematical formalism. Journal of Theoretical Biology 154 (4), 421–454.

[34] Schellenberger, J., Que, R., Fleming, R. M. T., Thiele, I., Orth, J. D., Feist, A. M., Zielinski, D. C., Bordbar, A., Lewis, N. E., Rahmanian, S., Kang, J., Hyduke, D. R., Palsson, B. Ø., 2011. Quantitative prediction of cellular metabolism with constraint-based models: the COBRA Toolbox v2.0. Nature Protocols 6 (9), 1290–1307.

[35] Slininger, P. J., Dien, B. S., Lomont, J. M., Bothast, R. J., Ladisch, M. R., Okos, M. R., 2014. Evaluation of a kinetic model for computer simulation of growth and fermentation by *Scheffersomyces* (*Pichia*) *stipitis* fed D -xylose. Biotechnology and Bioengineering 111 (8), 1532–1540.

[36] Vargas, F. A., Pizarro, F., Pérez-Correa, J. R., Agosin, E., 2011. Expanding a dynamic flux balance model of yeast fermentation to genome-scale. BMC Systems Biology 5 (75), 1–12.

[37] Varma, A., Palsson, B. O., Oct. 1994. Metabolic flux balancing: Basic concepts, scientific and practical use. Bio/Technology 12, 994–998.

[38] Vassiliadis, V. S., Sargent, R. W. H., Pantelides, C. C., 1994. Solution of a class of multistage dynamic optimization problems. 1. problems without path constraints. Industrial & Engineering Chemistry Research 33 (9), 2111–2122.

[39] Vassiliadis, V. S., Sargent, R. W. H., Pantelides, C. C., 1994. Solution of a class of multistage dynamic optimization problems. 2. problems with path constraints. Industrial & Engineering Chemistry Research 33 (9), 2123–2133.

[40] Xu, C., Liu, L., Zhang, Z., Jin, D., Qiu, J., Chen, M., 2013. Genome-scale metabolic model in guiding metabolic engineering of microbial improvement. Applied Microbiology and Biotechnology 97 (2), 519–539.

[41] Zhao, X., Noack, S., Wiechert, W., Lieres, E. V., 2017. Dynamic flux balance analysis with nonlinear objective function. Journal of Mathematical Biology, 1–29.

[42] Zomorrodi, A. R., Maranas, C. D., 2012. OptCom: A multi-level optimization framework for the metabolic modeling and analysis of microbial communities. PLoS Computational Biology 8 (2), e1002363.