An interior point solver for smooth convex optimization with an application to environmental-energy-economic models

Olivier Epelly*†, Jacek Gondzio**, Jean-Philippe Vial*

*Logilab, HEC, University of Geneva
102 Bd. Carl-Vogt, 1211-Geneva, Switzerland

†Systems Analysis Section, General Energy Research Department, Paul Scherrer Institute,
5232-Villigen, Switzerland

**Department of Mathematics and Statistics
University of Edinburgh, King's Buildings, Mayfield Road, Edinburgh EH9 3JZ, Scotland

July, 2000

Abstract

Solving large-scale optimization economic models such as MARKAL-MACRO models proves to be difficult or even out of reach for state-of-the-art solvers. We propose an optimizer which takes advantage of their possible special structure: a large dynamic linear block on one side, a small nonlinear convex block on the other one. This framework favors the use of interior point methods which are efficient for large-scale linear programs and which can handle convex programs. NLPHOPDM is an implementation of an interior point method built upon the HOPDM code for linear and convex quadratic optimization [18]. The method combines ideas of a globally convergent algorithm [3, 31] and the extension of multiple centrality correctors technique [19] to nonlinear convex programming. It is designed for being hooked to modeling languages such as GAMS and AMPL.
We present in this paper preliminary results relative to our research code NLPHOPDM and to commercial nonlinear solvers. Our approach achieves a significant computational speed-up. This is performed via the use of a library which computes exact second derivatives.
(*Interior Point Method; Economic Model; Smooth Convex Optimization*)

1 Introduction

MARKAL-MACRO [24] is a regional planning model for the energy sector that is frequently used to support energy policy assessment in climate change analysis. The model is formulated as a mathematical programming problem, involving a large set of linear constraints and a smaller component of smooth convex constraints. The main users are economists. They are mostly interested in scenario analyses; sometimes they also embed the model in a larger framework involving several countries to trace the economic impact of climate change. The tool they need and favor is an algebraic modeling language that makes it easy to define equations and modify data, and which is linked to an efficient and robust optimizer.

Presently, the existing MARKAL-MACRO models are all written in GAMS [7], a popular modeling language that is well-suited to the formulation of economic models. GAMS is linked to four general nonlinear solvers, namely CONOPT and CONOPT2 [11], MINOS [26] and SNOPT [17]. Currently, the row and column sizes of existing MARKAL-MACRO models range from $2,000 \times 3,000$ to $10,000 \times 15,000$, depending on the time mesh and the level of technological details. Even though the sizes are very reasonable by modern optimization standards, solving the larger MARKAL-MACRO problems on personal computers with the above software technology is still a challenge. This is particularly true when the model needs to be solved repeatedly, as it is the case when several independent MARKAL-MACRO models are components of a larger model solved by a decomposition scheme [5, 6, 8].

In this paper we present an alternative to the combinations GAMS/CONOPT or CONOPT2, GAMS/MINOS or GAMS/SNOPT. Namely, we have developed a new optimizer for smooth convex programming problems, and we have linked it to GAMS, to preserve ease of implementation for end-users. The algorithm requires the computation of exact first and second derivatives. Unfortunately, GAMS does not compute Hessian components, but other modeling languages, e.g., AMPL [15], offer this possibility. Since the users may not be eager to translate the GAMS models into AMPL, we propose a hybrid solution, mixing GAMS for the main linear part and AMPL for the more restricted nonlinear part. From the user point of view, the added complication of using AMPL to model the nonlinear part is very modest, all the more that the syntax in AMPL and GAMS is very similar.

The field of interior point methods for nonlinear programming has been rapidly growing in the recent past [2, 4, 9, 16, 30], giving rise to new solvers such as INTOPT [16], LOQO [30], MOSEK [2], NITRO [9] and OPINEL [4]. Our code may be viewed as one such proposal. NLPHOPDM is built upon the primal-dual interior point code HOPDM for linear and convex quadratic optimization [18] (`http://www.maths.ed.ac.uk/~gondzio/software/hopdm.html`) and extends the technique of [31]. It incorporates the linesearch of [31] and the higher-order predictor-corrector technique, initially implemented in [1, 19] for linear and convex quadratic problems. We name it NLPHOPDM thereafter, an acronym for Non-Linear Higher Order Primal-Dual Method. Our code is particularly well suited for problems with a large block of linear constraints. We exploit the two-components structure of MARKAL-MACRO models by differentiating perturbations of the linear and nonlinear components. This approach achieves a significant computational speed-up. MARKAL-MACRO models are currently solved by NLPHOPDM approximately from 3 to 5 times faster than by MINOS, from 4 to 70 times faster than by SNOPT, from 20 to 90 times faster than by CONOPT2 and from 50 to 370 times faster than by CONOPT. However, the code applies to more general smooth convex problems provided that it is supplied with second derivative values, an option that AMPL offers.

Our interior point implementation NLPHOPDM is briefly presented in section 2. MARKAL-

MACRO and related economic models are described in section 3. The linking of NLPHOPDM with modeling languages, namely AMPL and GAMS, is dealt with in section 4. Lastly, numerical results are presented in section 5, followed by our conclusions.

2 NLPHOPDM: an implementation of an interior point method

NLPHOPDM applies to problems of the following type

$$\begin{aligned} \min \quad & f(x) && \text{(1a)}\\ \text{s.t.} \quad & Ax = b, && \text{(1b)}\\ & g(x) \leq 0, && \text{(1c)}\\ & x \geq 0, && \text{(1d)} \end{aligned}$$

where $x \in \mathcal{R}^n,\ b \in \mathcal{R}^m,\ A \in \mathcal{R}^{m\times n},\ f : R^n \to R$ and $g : R^n \to R^l$. We assume that f and $g_i,\ i = 1, \ldots l$, are convex twice continuously differentiable functions. We further assume that the problem has a solution and that a constraint qualification holds at the optimum.

NLPHOPDM is a primal-dual infeasible interior point method for nonlinear convex programs. It aims at solving the first order optimality conditions:

$$\begin{aligned} Ax &= b, && \text{(2a)}\\ g(x) &\leq 0, && \text{(2b)}\\ \nabla f(x) + A^T w + \left(\frac{\partial g}{\partial x}(x)\right)^T y &\geq 0, && \text{(2c)}\\ X(\nabla f(x) + A^T w + \left(\frac{\partial g}{\partial x}(x)\right)^T y) &= 0, \quad x \geq 0, && \text{(2d)}\\ Yg(x) &= 0, \quad y \geq 0, && \text{(2e)} \end{aligned}$$

where $w \in \mathcal{R}^m$ and $y \in \mathcal{R}^l$ are Lagrange multipliers associated with linear and nonlinear constraints, respectively. X and Y are diagonal matrices with main diagonals x and y, respectively.
Actually, NLPHOPDM introduces slack variables x and z and aims at solving the following system of equations:

$$\begin{aligned} Ax &= b, && \text{(3a)}\\ g(x) + s &= 0, && \text{(3b)}\\ \nabla f(x) + A^T w + \left(\frac{\partial g}{\partial x}(x)\right)^T y - z &= 0, && \text{(3c)}\\ Xz &= 0, \quad x \geq 0,\ z \geq 0, && \text{(3d)}\\ Ys &= 0, \quad y \geq 0,\ s \geq 0. && \text{(3e)} \end{aligned}$$

For this purpose, NLPHOPDM tracks the solution of a perturbed system, in which (3a), (3d) and (3e) are replaced by the equations

$$\begin{aligned} Ax &= b + r, && \text{(4a)}\\ Xz &= \mu^1, \quad x \geq 0,\ z \geq 0, && \text{(4b)}\\ Ys &= \mu^2, \quad y \geq 0,\ s \geq 0. && \text{(4c)} \end{aligned}$$

There $r \in R^m$ is a perturbation, and $\mu^1 \in R^n$ and $\mu^2 \in R^l$ are positive vectors. Conditions (4a)–(4c), together with (3b)–(3c), form the first order optimality condition of the perturbed barrier problem

$$
\begin{aligned}
(P_{\mu,\, r}) \qquad \min \quad & f(x) - \sum_{i=1}^{n} \mu_i^1 \ \ln x_i - \sum_{j=1}^{l} \mu_j^2 \ \ln s_j && \text{(5a)}\\
\text{s.t.} \quad & Ax = b + r, && \text{(5b)}\\
& g(x) + s = 0, && \text{(5c)}\\
& x > 0,\ s > 0.
\end{aligned}
$$

NLPHOPDM drives the parameters r, μ_1, μ_2 to zero to yield an approximate solution of (3) of the desired precision. In classical barrier methods [13], one has $r = 0$, and the vectors μ^j, $j = 1, 2$ have equal components.

The basic step computes the Newton direction associated with (3b)–(3c) and (4). At the current iterate (x^k, s^k) in the primal space and (y^k, z^k, w^k) in the dual space, NLPHOPDM solves the linear system

$$
\begin{bmatrix}
A & 0 & 0 & 0 & 0\\
\frac{\partial g}{\partial x} & I & 0 & 0 & 0\\
H & 0 & \frac{\partial g}{\partial x}^T & -I & A^T\\
Z^k & 0 & 0 & X^k & 0\\
0 & Y^k & S^k & 0 & 0
\end{bmatrix}
\begin{bmatrix}
\Delta x^{k+1}\\ \Delta s^{k+1}\\ \Delta y^{k+1}\\ \Delta z^{k+1}\\ \Delta w^{k+1}
\end{bmatrix}
=
\begin{bmatrix}
b + r - Ax^k\\
-s^k - g(x^k)\\
-\nabla f(x^k) - A^T w^k - \frac{\partial g}{\partial x}^T y^k + z^k\\
\mu^1 - X^k z^k\\
\mu^2 - Y^k s^k
\end{bmatrix}, \qquad (6)
$$

where the first derivatives $\frac{\partial g}{\partial x}$ and second derivatives $H = \nabla^2 f(x) + \sum y_i \nabla^2 g_i(x)$ are evaluated at $x = x^k$ and $y = y^k$. The symbols S^k, X^k, Y^k and Z^k represent diagonal matrices built on corresponding vectors.

The algorithm can be shortly paraphrased as follows.

```
1- Check optimality within the specified tolerances.

2- Set new target values for r and μ.

3- Solve the Newton system associated with these targets.

4- Perform a damped Newton step along the Newton direction with possibly
different step lengths in the primal and dual spaces.
```

The main features of NLPHOPDM are the following.

- The system (6) is dynamically scaled to cope with poorly scaled formulation.
- The system (6) is reduced to a quasi-definite system and is solved using the Cholesky-like factorization of [1]. The routine exploits sparsity and applies a primal-dual regularization scheme to cope with ill-conditioning.
- The perturbation vector r in the early iterations is such that $Ax^k = b + r^k$, making the current iterate feasible for the primal linear constraint. Thus, the algorithm focuses on solving the nonlinear equations in the beginning. One should notice that once $r^k = 0$, i.e., x^k is primal feasible, then the search direction preserves the feasibility of the primal linear constraints.

- The vector parameters μ^j, $j = 1, 2$, are changed according to a strategy that proved successful in linear and quadratic programming [19]. Its theoretical background goes back to a target-following strategy [20]. The target-following strategy is extended to handle the parameter r in the linear equations.
- The iterative step is of the predictor-corrector type [25], with the multiple corrections technique of [19].

NLPHOPDM is well-suited to handle convex problems such as MARKAL-MACRO that mix many sparse linear constraints and smooth convex constraints. The convergence result of [31] was studied in [3]. The algorithm we propose relies on these developments. NLPHOPDM can be accessed directly through AMPL or a combination of GAMS and AMPL, which makes its use possible by nonspecialists.

3 Application to environmental-energy-economic models

NLPHOPDM was first developed with the aim of solving MARKAL-MACRO [24], an environmental-energy-economic model currently used at the Paul Scherrer Institute.

MARKAL-MACRO links together a model derived from Ramsey macroeconomic growth model [28], called MACRO, and MARKAL [14], a general energy model developed in the late 70's in the framework of the Environmental Technology System Analysis Program, a project initiated by the International Energy Agency. MARKAL-MACRO is used to analyze the coupling between economic growth, the level of energy demands and the evolution of an energy system to supply these demands. MARKAL-MACRO is nowadays used as a common kernel in many countries around the world, each country setting up its own database and model refinements. This section briefly presents MARKAL-MACRO. It also presents a discretized version of the Ramsey model and a 2-goods MERGE model [23], which are economic models closely related to MARKAL-MACRO and which are interesting from the benchmarking point of view. Numerical results are given for all three models in section 5.

3.1 MARKAL-MACRO

Next subsections present MARKAL-MACRO model definition (3.1.1) as well as different formulations with the help of a modeling language and the associated standard solution algorithms (3.1.2).

3.1.1 Model definition

In MARKAL-MACRO, a single agent is representative of all economic agents; this agent has perfect foresight and is motivated by the maximization of a welfare function, which is an additive function in consumptions.
Mathematically, MARKAL-MACRO is expressed as the following dynamic optimization problem over T periods:

$$\max_{C,I,K,D,EC,x\geq 0} U = \sum_{t=1}^{T} \beta_t \log C_t$$

$$\text{subject to: } \forall\, t \in \{1, \ldots, T\}$$

$$[\mathrm{a} K_t^{\rho\alpha} L_t^{\rho(1-\alpha)} + \sum \mathrm{b}_m D_{mt}^{\rho}]^{1/\rho} = C_t + I_t + EC_t, \tag{7a}$$

$$
\begin{aligned}
K_{t+1} &= \mathrm{c}K_t + \mathrm{d}I_t + \mathrm{e}I_{t+1}, && \text{(7b)}\\
\mathrm{A}_t x_t &= D_t, && \text{(7c)}\\
\mathrm{M}_t x_t &= EC_t, && \text{(7d)}\\
\mathrm{F}_t x_t &= 0, && \text{(7e)}\\
\mathrm{E}_t x_t &\leq \mathrm{B}_t, && \text{(7f)}\\
\mathrm{U}_t x_t &\leq \mathrm{R}_t, && \text{(7g)}\\
\text{and to: } I_T &\geq \mathrm{f}K_T. && \text{(7h)}
\end{aligned}
$$

where :

- variables C_t, K_t and I_t are respectively macroeconomic consumption, capital and investment at period t;
- variable EC_t stands for energy costs at period t and variables D_{mt} stand for the corresponding energy demand for each sector m;
- variables x_t are investments and activity levels in the energy sector at period t;
- the aggregated output of economy is a concave function in capital, labour and energy services, which is invested or consumed, the energy part of those investments and consumptions being accounted for separately (constraints (7a));
- there is an intertemporal accumulation of capital (constraints (7b));
- activities $A_t x_t$ in the energy sector meet energy demand (constraints (7c));
- cost $M_t x_t$ of investments and activities in the energy sector at period t is accounted for in EC_t (constraints (7d));
- flows $F_t x_t$ of investments and energy fuels are balanced (constraints (7e));
- emissions $E_t x_t$ induced by activities in the energy sector are upper bounded by environmental constraints (constraints (7f));
- use $U_t x_t$ of resources by activities in the energy sector cannot exceed available resources (constraints (7g));
- infinite time horizon is approximated by a lower bound over last period investment (constraint (7h)).

3.1.2 Formulation and standard solution algorithms

Variables EC_t and D_{mt} link together the nonlinear submodel MACRO and the linear submodel MARKAL. Typically, the linear part consists of several thousands of constraints, whereas the nonlinear part involves between 5 and 10 constraints, depending on the time mesh.

MARKAL-MACRO can be reformulated as a *nonlinearly constrained convex program* by changing nonlinear constraints (7a) as follows:

$$
[\mathrm{a}K_t^{\rho\alpha} L_t^{\rho(1-\alpha)} + \sum \mathrm{b}_m D_{mt}^{\rho}]^{1/\rho} \geq C_t + I_t + EC_t. \qquad (8)
$$

This reformulation yields the same optimum as the original formulation (7) because any solution with a strictly positive slack cannot be optimal, since it could be allocated to variable C_t and thus the objective function would be improved.

MARKAL-MACRO can also be formulated as a *linearly constrained convex program* just by substituting variables C_t involved in constraints (7a) for variables C_t involved in the objective function in this way:

$$\max_{I,K,D,EC,x\geq 0} \sum_{t=1}^{T} \beta_t \log(\ [\mathrm{a} K_t^{\rho\alpha} L_t^{\rho(1-\alpha)} + \sum \mathrm{b}_m D_{mt}^{\rho}]^{1/\rho} - I_t - EC_t). \tag{9}$$

Note that this latter formulation imposes that the arguments in the logarithms in the objective function be positive, though no constraints expresses it explicitly. This is incompatible with the use of NLPHOPDM because the algorithm used in it requires such implicit constraints to be explicit.

Summing up, MARKAL-MACRO can be formulated as a convex program, linearly constrained or not. Besides, it can be shown that constraints qualification holds at optimality [8]. In the end, all the convexity and constraint qualification assumptions required for the use of NLPHOPDM are satisfied.

MARKAL-MACRO is written in GAMS [7] algebraic modeling language. GAMS is a very flexible general-purpose language which enables the modeling of a variety of economic and engineering problems. Implementation of large models such as MARKAL-MACRO is not tractable unless such a modeling language is used.

On the solution side, as far as nonlinear optimization is concerned, GAMS is hooked to four solvers: CONOPT and CONOPT2 [11], MINOS [26] and SNOPT [17]. For problems with nonlinear constraints, MINOS uses a projected Lagrangian method, whereas problems with linear constraints are solved by a reduced-gradient method with quasi-Newton approximations to the reduced Hessian. SNOPT employs a sparse sequential quadratic programming algorithm with limited-memory quasi-Newton approximations to the Hessian of Lagrangian. If the constraints are highly nonlinear, or the functions and gradients are expensive to evaluate, then SNOPT may be more effective than MINOS. CONOPT and CONOPT2 are feasible path methods based on the generalized reduced gradient algorithm. They are particularly well suited for very nonlinear models with few degrees of freedom, i.e., the number of variables is approximately the same as the number of constraints, which is not the case of MARKAL-MACRO.

MARKAL-MACRO users usually choose MINOS rather than CONOPT, CONOPT2 or SNOPT because MARKAL-MACRO can be formulated as a linearly constrained program that MINOS handles efficiently. Besides, MINOS is all the more efficient that only some of the variables enter nonlinearly, as it is the case in MARKAL-MACRO. Experience shows that the algorithms behavior strongly depends on the problem instance, sometimes in an unpredictable way. For this reason, the users appreciate having more than one solver at their disposal, so that if one fails, there are still chances that another one succeeds. In this respect, NLPHOPDM adds to the existing collection of available software. From a practical point of view, the use of NLPHOPDM requires the nonlinearly constrained convex formulation (8) of MARKAL-MACRO to be used, and a code which computes second derivatives must be supplied. Section 4 shows how this can be achieved in a modest amount of work.

3.2 Ramsey model

A version of the discrete model studied by Ramsey [28] is as follows:

$$\text{maximize} \sum_{t=1}^{T-1} \beta^t \log C_t + \frac{\beta^T}{1-\beta} \log C_T$$

subject to:

$$\begin{aligned} \alpha L_t^{1-b} K_t^b &= C_t + I_t \quad \forall\, t, && (10a)\\ K_{t+1} &\leq K_t + I_t \quad \forall\, t > 1, && (10b)\\ I_T &\geq \eta\, K_T. && (10c) \end{aligned}$$

Variables are macroeconomic consumption C_t, capital K_t and investment I_t for each period t; Labour L_t s an exogenous parameter. A convex formulation can be obtained by replacing the equality in (10a) by an appropriate inequality.

This model is interesting in that MARKAL-MACRO macroeconomic submodel is based on it. But in contrast to MARKAL-MACRO, the Ramsey model includes as many linear constraints as nonlinear constraints. However, a linearly constrained convex formulation can be obtained by eliminating variables C_t.

3.3 SMERGE

MERGE [23] is a multi-regional optimization problem formulated as a Negishi-welfare problem. In SMERGE (standing for Small MERGE), two goods g, namely numeraire, denoted num, and carbon emission rights, denoted crt, are exchanged by regions $r = 1, \ldots, R$ over $t = 1, \ldots, T$ periods. The model is formulated as follows:

$$\max \sum_{r=1}^{R} \alpha_r \sum_t \log C_{r,t}$$

subject to:

$$\begin{aligned} \sum_r X_{r,g,t} &= 0 \quad \forall\, t \ \forall\, g, && (11a)\\ Z_{r,t} + X_{r,crt,t} &= \text{share}_{r,t} \times ZG_t \quad \forall\, r \ \forall\, t, && (11b)\\ ZC_t &= ZC_{t-1} + 0.5 \times \text{nyper} \times (ZG_{t-1} + ZG_t) \quad \forall\, t > 1, && (11c)\\ K_{r,t} &= \text{ksur}_{r,t} \times K_{r,t-1} + 0.5 \times (\text{ksur}_{r,t-1} + I_{r,t-1}) + I_{r,t} \quad \forall\, r \ \forall\, t > 1, && (11d)\\ I_{r,T} &\geq (\text{grow}_{r,T} + \text{depr}_r) \times K_{r,T} \quad \forall\, r, && (11e)\\ Y_{r,t} &= \text{a}_r \times K_{r,t}^{\text{kpvs}_r} \times L_{r,t}^{\text{lpvs}_r} \quad \forall\, r \ \forall\, t, && (11f)\\ Y_{r,t} &= C_{r,t} + I_{r,t} + X_{r,num,t} + \text{acq}_{r,t} \times AQ_{r,t}^2 + \text{acb}_{r,t} \times AB_{r,t} \quad \forall\, r \ \forall\, t, && (11g)\\ AB_{r,t} &\leq 0.01 \times \text{zbase}_{r,t} + 2 \times AB_{r,t-1} \quad \forall\, r \ \forall\, t > 1, && (11h)\\ Z_{r,t} &= \text{zbase}_{r,t} - AQ_{r,t} - AB_{r,t} \quad \forall\, r \ \forall\, t. && (11i) \end{aligned}$$

At each iteration of the so-called Negishi loop, coefficients α_r of regional utilities are updated by the following formula which involves primal and dual solution λ of the current formulation:

$$\alpha_r := \sum_t \frac{\lambda(11a)_{t,num}}{\lambda(11a)_{T,num}} \times C_{r,t} + \sum_{t,g} \frac{\lambda(11a)_{t,g}}{\lambda(11a)_{T,num}} \times X_{r,g,t}.$$

Variables of the model are as follows:

- K, C and I, defined in the same way as in MARKAL-MACRO, Y, conventional gross national product and X, net exports of tradeable goods;
- AQ, quadratic abatement level, and AB, backstop abatement level;
- Z, ZG and ZC, which stand for regional, global and global cumulative CO_2 emissions.

Trade balance equations (11a), global emissions sharing equations (11b) and cumulative global carbon emissions equations (11c) are central constraints. Equations (11d)-(11i) are regional constraints. A nonlinearly convex formulation of SMERGE can be obtained by replacing the equalities in (11f) and (11g) by appropriate inequalities. A linearly constrained convex formulation can be obtained by eliminating variables C_t.

4 Interface with modeling languages

NLPHOPDM is designed to be hooked to a modeling environment. The interest in linking solvers to modeling languages is that solvers can be used by nonspecialist in optimization. Subsection 4.1 describes the basic interface between NLPHOPDM and AMPL. The interested reader can refer to [12] for a detailed description. Subsection 4.2 presents the hybrid interface with GAMS and AMPL that feeds NLPHOPDM with the first and second order derivatives.

4.1 Basic interface

Starting from a model formulation, a modeling language generates a formatted output file which can be handled by an optimization code. The translation of the model formulation into a formatted output depends upon whether equations are linear or not. Linear equations are translated once and for all into a matrix containing constant coefficients. In most economic models, the matrix is usually sparse; it is stored in a format which is handled efficiently by solution algorithms. Nonlinear equations are processed so that routines included in the modeling language software can efficiently compute values and derivatives of equations at a given point. During the solve process, these routines are iteratively called by solution algorithms in due course of time. NLPHOPDM is hooked to AMPL in this way, because this modeling environment provides routines which compute first *and* second derivatives, both being used in NLPHOPDM algorithm, as it appears in Figure 1.

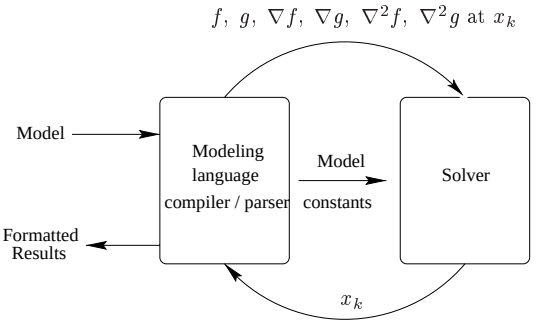

Figure 1: Basic interface.

4.2 Hybrid interface

Since GAMS cannot compute second derivatives, the basic interface between NLPHOPDM and GAMS cannot be used stand alone to solve a model formulated in GAMS. Well-known techniques exist in order to approximate second derivatives from first derivatives, but this leads to a less efficient behavior of algorithms. If such a technique is useful in the case of nonanalytic functions, an analytical solution is preferable in the case of MARKAL-MACRO since the model equations are explicitly formulated. However, developing computer routines for computing exact second derivatives would be far too demanding on the users: this task is time-consuming and prone to errors. From the modeler point of view, it is clearly much more convenient and efficient to remain within their preferred modeling environment as usual, and to use routines already developed for this purpose.
It is an easy task to write the nonlinear equations (7) in AMPL, and retrieve from this library the second order information. The challenge is to have the two environments work together without imposing much extra work on the user, which is allowed by the hybrid interface described in Figure 2:

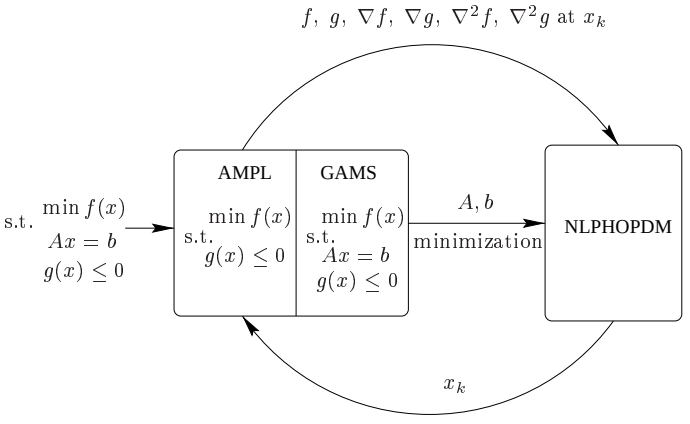

Figure 2: Hybrid interface.

5 Numerical results

In this section, problems presented in section 3 are used to compare CONOPT, CONOPT2, MINOS, NLPHOPDM and SNOPT in terms of runtime. All problems written in GAMS modeling language are run by NLPHOPDM *via* the hybrid interface with AMPL described in section 4. We also compare NLPHOPDM and LOQO in terms of iterations for problems written in AMPL.

5.1 Problems characteristics

Models under study are Ramsey, SMERGE and MARKAL-MACRO models:

- RAM200 and RAM400 denote Ramsey models for respectively 200 and 400 time periods;
- SMERGE denotes a -2 goods, 5 regions, 7 time periods- version of MERGE [23];

- MM-CH5, MM-CH9, MM-COL, MM-SWE and MM-US denote MARKAL-MACRO models respectively for Switzerland [6, 21, 22] over 5 and 9 time periods, for Colombia [10] over 7 time periods, for Sweden [27] over 9 time periods and for US [29] over 8 time periods. All MARKAL-MACRO models are based on the same set of equations (except for Colombia which slightly departs from it) but rely on different databases.

Problems sizes are not only proportional to the number of time periods but also depend on the description level of the economy under study. Problems sizes range from small to medium and can be described by the number of nonzero elements, denoted nzJ, in the Jacobian matrix:

Problems	rows	columns	nzJ
RAM200	400	600	1,400
RAM400	800	1,200	2,800
SMERGE	450	600	1,600
MM-CH5	2,000	3,000	22,000
MM-CH9	3,500	5,500	44,000
MM-COL	3,400	5,200	34,000
MM-SWE	5,400	7,200	62,000
MM-US	10,200	15,800	125,000

Table 1: MARKAL-MACRO models characteristics.

5.2 Solution platform

Except for the comparison with LOQO 5.02, all runs were performed on a 350 MHz Pentium II processor under Linux, with 96 MB of RAM, a level-1 memory cache of 32 kB and a level-2 memory cache of 512 kB. The executable of LOQO 5.02 that we downloaded (`http://www.orfe.princeton.edu/~loqo`) was apparently not compatible with the C library of this machine and was therefore compared with NLPHOPDM on another compatible machine under Linux running at 200 MHz. Last versions of softwares, and possibly previous ones, were used, i.e.,

- CONOPT 2.043C and CONOPT2 2.071C;
- LOQO 4.01 and LOQO 5.02;
- MINOS 5.4 and MINOS 5.5;
- NLPHOPDM 1.0;
- SNOPT 5.3.5.

Comparison between NLPHOPDM and LOQO is done both on the basis of runtime and in terms of iterations number, since both optimizers are path-following primal-dual methods. The problems submitted to LOQO are all written in AMPL because LOQO cannot be used to solve models written in GAMS. LOQO can solve nonconvex programs but it has an option for convex cases which was turned on for fair comparison with NLPHOPDM. LOQO also has an option for turning on the predictor-corrector technique. Runs were done with and without this option and only the best results are presented.

5.3 Algorithms setting

Each algorithm was run with its default setting, excepted possibly as far as optimality criteria are concerned (see 5.4). Adjustments were made if one of the following event occurred:

- The maximum number of iterations allowed was reached. This occurred for some runs with CONOPT, CONOPT2, MINOS and SNOPT. The new setting consisted in increasing the iterations number limit.
- The time limit was reached. This occurred for some runs with CONOPT, CONOPT2, MINOS and SNOPT. The new setting consisted in increasing the time limit.
- A Jacobian element or a variable had exceeded a given threshold. This was necessary for some runs with CONOPT and CONOPT2. The new setting consisted in increasing those thresholds.

Last, CONOPT, CONOPT2, MINOS and SNOPT sometimes stopped because there was no change in the objective after a given number of iterations; such a situation is well known by MARKAL-MACRO users and it is managed at the level of the modeling language by a loop which consists of an iterative solving with a *warm* start. In MARKAL-MACRO models, the number of attempts is usually 3 and it was increased up to 7 to increase chances of reaching optimality.

5.4 Stopping criteria

A fair comparison can be performed provided that algorithms have the same stopping criteria and the same tolerances. All solvers under examination, except LOQO and CONOPT, run with 1E-6 as default tolerance. We accordingly changed LOQO tolerance, but let CONOPT default tolerances, because this latter optimizes the choice of the tolerance level by taking into account the value of the machine-dependent finite precision and by adjusting dynamically the tolerance level in the course of the algorithm run. In the end, for the problems under examination, CONOPT stopped at optimal solutions for which stopping criteria were close to those of other solvers. The different criteria types are summarized below for the different optimizers as they are reported at `http://www1.gams.com/docs/pdffiles.htm` for CONOPT and CONOPT2 and at `http://www.sbsi-sol-optimize.com/manuals.htm` for MINOS and SNOPT.

CONOPT default stopping criteria:

- Feasibility tolerance: a constraint is considered feasible if the residual is less than 1E-3 $\times \frac{MaxJac}{100}$, where `MaxJac` is an overall scaling measure for the constraints, computed as $\max(\max_{ij} J_{ij}, 100)$, where `J` denotes the Jacobian matrix.
- Optimality tolerance: its value depends on machines and is usually around 9E-8; optimality is reached if the largest superbasic component of the reduced gradient is lower than this tolerance.
- Information displayed at optimality: the largest reduced gradient of the nonoptimal variables (denoted `RGmax`).

CONOPT2 default stopping criteria:

- CONOPT2 uses the same criteria as CONOPT.

- Information displayed at optimality: the largest reduced gradient of the nonoptimal variables (denoted `RGmax`).

LOQO default stopping criteria:

- Feasibility tolerance: 1E-6 is the default tolerance for primal and dual relative feasibility tolerances.
- Optimality tolerance: 1E-8 is the default tolerance for the absolute gap between primal and dual objectives. In view a comparing LOQO with NLPHOPDM as fairly as possible, we set LOQO optimality tolerance to an appropriate problem-dependent value since NLPHOPDM uses a relative optimality tolerance equal to 1E-6.
- Information displayed at optimality: primal and dual feasibility violations (`Pviol` and `Dviol`) and primal and dual objective values at optimality (`Pobj` and `Dobj`).

MINOS 5.4 default stopping criteria:

- Feasibility tolerance: 1E-6 is the default feasibility tolerance for linear constraints.
- Row tolerance: 1E-6 is the default relative feasibility tolerance for nonlinear constraints, the latter being scaled by a measure of the size of the current primal solution.
- Optimality tolerance: 1E-6 is the default relative tolerance for the relative norm (denoted `RG/PI`) of the reduced gradient of linearized subproblems, scaled by $\max(\frac{\sum_{i=1}^{m} |\pi_i|}{\sqrt{m}}, 1)$, where π denotes the dual variables.
- Information displayed at optimality: the maximum component of the nonlinear constraint residuals (denoted `Viol`), the largest relative constraint violation, scaled by a a measure of the size of the current primal solution (denoted `Viol/X`). the norm of the reduced gradient of the last linearized subproblem (denoted `RG`) and the same norm scaled by the above norm of dual variables (denoted `RG/PI`).

MINOS 5.5 default stopping criteria:

- Feasibility tolerance: 1E-6.
- Optimality tolerance: 1E-6.
- Information displayed at optimality: a measure of infeasibility (denoted `Feasible`) and a measure of optimality (denoted `Optimal`) in the case of nonlinearly constrained problems, two measures of infeasibility in the case of linearly constrained problems (denoted `FeasSum` and `FeasMax`).

NLPHOPDM default stopping criteria:

- Feasibility tolerance: 1E-6 is the default tolerance for primal and dual constraints relative feasibility. Primal residual i is scaled by $\max(1, |b_i|)$, dual residual k is scaled by $\max(1, |\nabla_k f(x)|)$, where f is the objective function and x the current primal variables. Note that we depart from most criteria used for feasibility. Usually, each residual is scaled by a global measure of coefficients magnitude, i.e., primal variables norm for primal feasibility, dual variables norm for dual feasibility, or right-hand side norm of given

primal or dual space. This option is well justified in LP where a preliminary scaling process sets coefficients around 1. In NLP, we find that a local measure of the relative accuracy of residuals is more appropriate.

- Optimality tolerance: 1E-6 is the default tolerance for the relative complementarity gap scaled by $\max(1, |f(x)|)$.
- Information displayed at optimality: the maximum components of relative violations of primal and dual constraints (denoted `relPviol` and `relDviol`) and the relative complementarity gap (denoted `relGap`). NLPHOPDM considers that optimality is reached when all stopping criteria are satisfied, either for formulation (2) or for formulation (3).

SNOPT default stopping criteria:

- Minor feasibility tolerance: 1E-6 is the default feasibility tolerance for linear constraints.
- Major feasibility tolerance: 1E-6 is the default relative feasibility tolerance for nonlinear constraints, the latter being scaled by the 2-norm of primal variables.
- Major optimality tolerance: 1E-6 is the default relative tolerance for the largest complementarity gap scaled by the 2-norm of dual variables.
- Minor optimality tolerance: 1E-6 is the tolerance for the norm of the reduced gradient of quadratic subproblems scaled by $\max(\frac{\sum_{i=1}^{m} |\pi_i|}{\sqrt{m}}, 1)$, where π denotes the dual variables.
- Information displayed at optimality: the maximum component of the scaled nonlinear constraint residuals (denoted `Feasible`) and the value of the maximum complementarity gap (denoted `Optimal`). In the case of linearly constrained problems, only `Optimal` is displayed.

It is clear from this overview that tolerances for nonlinear optimizers can be set identical but the way criteria are built differs from one optimizer to another. As a result, it is almost impossible to perform a comparison exactly on the same basis. The additional information delivered by optimizers at optimality allows to check, to some an extent, whether significant differences appear. For example, MINOS 5.5 and SNOPT claim that they solve the MM-US model, whereas some variables are reported as being nonoptimal. However, for the models we run, results can be considered as conclusive enough in view of a rough comparison of optimizers efficiency, for this class of problems. We refer the reader to detailed results in appendix which support this conclusion.

5.5 Comparison with CONOPT, CONOPT2, MINOS and SNOPT

Next tables indicate the solution time durations in case of successful runs. The letter "F", if present, indicates a failure of the corresponding optimizer. The symbol "-" is used if a formulation is inappropriate. For problems written in GAMS, execution time is measured by the `user time` returned by the `time` shell command, which thus records CPU seconds both for the optimizer and the modeling language itself. In the case of the AMPL version of SMERGE, execution time is measured by AMPL built-in `_total_solve_time` command.

Table 2 corresponds to nonlinearly constrained convex formulation of problems, Table 3 stands for linearly constrained convex formulations, Table 4 reports the best times in between

tables 2 and 3 and Table 5 reports time durations ratios over the best time among the solvers under examination.

Problems	NLPHOPDM	CONOPT	CONOPT2	MINOS 5.4	MINOS 5.5	SNOPT
RAM200	1.3	3.4	5.5	6.5	17.2	20
RAM400	38	9.4	40	15	97	61
SMERGE	4.3	21.3	F	15.5	7.3	145
MM-CH5	20	1,000	950	F	600	1,080
MM-CH9	85	11,240	7,920	4,000	5,750	22,400
MM-COL	240	23,220	5,180	4,500	7,330	4,350
MM-SWE	240	90,000	13,130	F	3,180	5,330
MM-US	480	33,000	43,000	F	1,330	2,000

Table 2: Computing times (sec.) for nonlinearly constrained convex formulations.

Problems	NLPHOPDM	CONOPT	CONOPT2	MINOS 5.4	MINOS 5.5	SNOPT
RAM200	-	0.84	1.86	2.76	2.70	5.55
RAM400	-	1.87	25.9	5.88	5.55	9.4
SMERGE	-	-	-	-	-	-
MM-CH5	-	-	870	90	70	820
MM-CH9	-	-	F	560	440	6,030
MM-COL	-	-	F	2,970	680	2,550
MM-SWE	-	-	F	1,200	700	2,660
MM-US	-	-	F	F	5,200	23,500

Table 3: Computing times (sec.) for linearly constrained convex formulations.

Problems	NLPHOPDM	CONOPT	CONOPT2	MINOS 5.4	MINOS 5.5	SNOPT
RAM200	1.32	0.84	1.86	2.76	2.70	5.55
RAM400	38	1.87	25.9	5.88	5.55	9.4
SMERGE	4.30	21.3	F	15.5	7.3	145
MM-CH5	20	1,000	870	90	70	820
MM-CH9	85	11,240	7,920	560	440	6,030
MM-COL	240	23,220	5,180	2,970	680	2,550
MM-SWE	240	89,000	13,130	1,200	700	2,660
MM-US	480	33,000	43,000	F	1,330	2,000

Table 4: Best computing times in seconds for all formulations.

NLPHOPDM achieves the best computing times for all models of this set except for Ramsey models. To be fair, let us point out that the solvers to which NLPHOPDM is compared are not restricted to *convex* programs. They can solve general nonlinear programs and therefore do not take advantage of convexity as NLPHOPDM does.

Problems	NLPHOPDM	CONOPT	CONOPT2	MINOS 5.4	MINOS 5.5	SNOPT
RAM200	1.6	1	2.2	3.3	3.2	6.6
RAM400	20	1	14	3	3	5
SMERGE	1	5	∞	3.6	1.7	34
MM-CH5	1	50	40	4	3	40
MM-CH9	1	130	90	6	5	70
MM-COL	1	100	20	10	3	10
MM-SWE	1	370	55	5	3	10
MM-US	1	70	90	∞	3	4

Table 5: Approximate ratios over best computing times.

MM-US is probably the most difficult problem to solve in this set. MINOS 5.5 and SNOPT, as NLPHOPDM, produce a solution that is claimed optimal. However, in their results file, both MINOS 5.5 and SNOPT report numerous nonoptimal variables and/or feasibility violations. Last, let us observe that CONOPT and CONOPT2 are, contrary to NLPHOPDM, specially designed to handle strongly nonlinear models. Their better behaviour for Ramsey models rather than for MARKAL-MACRO models is therefore consistent with their design.

5.6 Comparison with LOQO

LOQO is not linked with GAMS modeling environment, because this latter does not provide Hessian. As a result, comparison is restricted to those problems presented in section (3) which are formulated in AMPL too. Comparison is performed on the basis of the number of iterations as well as of the time required to reach optimality. For SMERGE, the sum of the iterations performed at each iteration of the Negishi loop is reported.

	NLPHOPDM		LOQO 4.01		NLPHOPDM		LOQO 5.02	
	runs on a 350 MHz machine				runs on a 200 MHz machine			
Problems	iter.	time	iter.	time	iter.	time	iter.	time
RAM200	30	1.3	3,410	1,040	30	2.3	58	2.67
RAM400	459	38	F	F	459	63	F	F
SMERGE	73	4.30	123	5.25	73	7.08	115	5.37

Table 6: Number of iterations and runtime (sec.).

5.7 Computational effort and problems size

Table 7 reports the number of iterations and the solution time needed by NLPHOPDM for MARKAL-MACRO models in relation with the size of the problems. In contrast to Table 1, size is given here after problems were preprocessed by NLPHOPDM, which implies the introduction of additional slack variables and also the possible removal of fixed variables and of redundant constraints. Size is expressed in terms of rows and columns numbers and of nonzeros in Jacobian (nzJ) but also in terms of nonzeros in Hessian (nzH) and in the matrix (6) to be factorized (nzTOTAL), because it is more directly related to the algorithm used in NLPHOPDM.

Problems	iterations	time (sec.)	rows	columns	nzJ	nzH	nzTOTAL
MM-CH5	31	20	2,000	4,800	15,000	1,600	33,000
MM-CH9	33	85	3,700	8,600	30,000	2,800	63,000
MM-SWE	47	240	4,700	11,000	39,000	1,600	80,000
MM-COL	54	240	3,200	7,600	28,000	31,000	87,000
MM-US	52	480	10,000	23,000	77,000	13,500	168,000

Table 7: Computational effort and problems size.

Problems are listed in increasing order of nzTOTAL. The computational effort in NLPHOPDM seems to be nicely related to problems size, whereas it does not hold so much for other solvers.

6 Conclusions

We propose an optimizer which takes advantage of the special structure of large-scale economic models such as MARKAL-MACRO models: a large dynamic linear program on one side, a small nonlinear convex block on the other one. This framework favors the use of interior point methods which are efficient for large-scale linear programming and which can handle convex programs. NLPHOPDM is an implementation of an interior point method built upon the HOPDM code for linear and convex quadratic programming [18]. Its algorithm combines ideas of a globally convergent algorithm [3, 31] and the extension of multiple centrality correctors [19] to nonlinear convex programming. It is designed for being hooked to modeling languages such as GAMS and AMPL. Additional information can be found at the following pages:
`http://www.maths.ed.ac.uk/~gondzio/software/hopdm.html` and,
`http://www.ecolu-info/~logilab/software/nlphopdm.html`.
In this paper, we reported preliminary numerical experiments with NLPHOPDM. The code is still under development. No special effort has been made to improve basic i/o operations nor to optimize numerical procedures. In this respect our results are particularly encouraging in comparison with other codes. Benchmarking was achieved as fairly as possible, based on the same tolerance level (six significant figures), even though it appears difficult to compare codes based on different methods, because stopping criteria are not always equivalent. This is particularly true for the larger models under examination. We observe that the "optimal" objective values reported by each optimizer differ by amounts larger than the announced optimality tolerances. However, we may fairly conclude that MARKAL-MACRO models are currently solved by NLPHOPDM approximately from 3 to 5 times faster than by MINOS, from 4 to 70 times faster than by SNOPT, from 20 to 90 times faster than by CONOPT2 and from 50 to 370 times faster than by CONOPT. This is performed via the use of a library which computes exact second derivatives.
Among obvious developments liable to reduce computing time, we would like to point out that the management of first and second order derivatives can be improved greatly; right now, we build a whole Jacobian matrix at each iteration, instead of just updating nonlinear coefficients. Further applications will concern very large-scale nonlinear convex programs which can be solved by decomposition techniques. It is expected that NLPHOPDM will speed up the solution process, since each local model must be repeatedly solved after a slight perturbation.

7 Acknowledgements

We thank Angela Cadena, Socrates Kypreos, Alan Manne, Ingrid Nyström and Philipp Tseng for making available their models.

8 Appendix

8.1 Detailed results

Next tables report detailed outputs of all runs. The following status is indicated:

- Optimal: whenever the solver converges within prespecified tolerance, the corresponding stopping criteria and objective function values are reported. Only significant figures are reported as much as possible. Note that it is not always clear whether a figure is significant or not.
- F: whenever the optimum is not found, we give corresponding details as much as possible.

8.2 CONOPT

For the nonlinearly constrained formulation of SMERGE, CONOPT claims that it found an optimal solution but `RGmax` is equal to 1.2E-02. For linearly constrained formulations of MARKAL-MACRO and MERGE, CONOPT reports that too many domain errors occurred in nonlinear functions, due to arguments of logarithmic terms in the objective function which became negative.

Problems	RGmax	Obj. value	Status
RAM200	1.8E-05	9.35021	Optimal
RAM400	1.7E-05	9.34818	Optimal
SMERGE	1.2E-02	2,327,920.7	Optimal
MM-CH5	3.2E-08	2,802.6951	Optimal
MM-CH9	1.9E-05	5,248.7	Optimal
MM-COL	1.6E-04	4,204.6	Optimal
MM-SWE	5.6E-04	12,504.6	Optimal
MM-US	1.8E-05	7,862.1	Optimal

Table 8: CONOPT results for nonlinearly constrained convex formulations.

Problems	RGmax	Obj. value .	Status
RAM200	9.4E-06	9.3522621	Optimal
RAM400	1.6E-05	9.3524809	Optimal
SMERGE			F
MM-CH5			F
MM-CH9			F
MM-COL			F
MM-SWE			F
MM-US			F

Table 9: CONOPT results for linearly constrained convex formulations.

8.3 CONOPT2

For the nonlinearly constrained convex formulation of SMERGE, CONOPT2 reports that it found a feasible solution after 21,6 seconds but that convergence was too slow. The objective value was equal to 2,327,920.6 at the last iterate, which is close to the optimal value, but `RGmax` is equal to 2.3E-01. For the linearly constrained convex formulation of SMERGE, CONOPT2 stopped after a too high number of errors during the evaluation of the objective function. For linearly constrained convex formulations of MM-CH9, MM-COL, MM-SWE and MM-US, CONOPT2 reports after respectively 3,300, 2,600, 5,800 and 34,000 seconds that it found a feasible solution but that there is no change in objective although the reduced gradient is greater than the tolerance. Note that the point at which CONOPT2 stopped for MM-US is relatively close to optimality since `RGmax` is equal to 7.8-04 and the objective value is equal to 7,861.

Problems	`RGmax`	`Obj. value`	`Status`
RAM200	7.6E-08	9.3536537	Optimal
RAM400	6.3E-08	9.3538929	Optimal
SMERGE			F
MM-CH5	3.7E-08	2,802.6952	Optimal
MM-CH9	6.4E-08	5,249.1968	Optimal
MM-COL	3.2E-08	4,205.8056	Optimal
MM-SWE	2.2E-08	12,535.996	Optimal
MM-US	1.3E-06	7,867.14	Optimal

Table 10: CONOPT2 results for nonlinearly constrained convex formulations.

Problems	`RGmax`	`Obj. value`	`Status`
RAM200	2.4E-08	9.3536548	Optimal
RAM400	8.1E-08	9.3539487	Optimal
SMERGE			F
MM-CH5	2.1E-08	2,802.6951	Optimal
MM-CH9			F
MM-COL			F
MM-SWE			F
MM-US			F

Table 11: CONOPT2 results for linearly constrained convex formulations.

8.4 MINOS 5.4

Failures reported in Table 12 correspond to two different situations: in MM-CH5 case, we interrupted the solver after 2,000 seconds because erratic iterates were generated, whereas in the case of MM-SWE and MM-US, MINOS 5.4 stopped because the current point, far from the optimal solution, could not be improved upon. In Table 13, for the MM-US model, MINOS 5.4 stopped after 4,000 seconds for the same reason, whereas the point at which MINOS 5.4 stopped is relatively close to optimality since RG and RG/PI are equal to 4.4E-02 and the objective value is equal to 7,865. For the linearly constrained convex formulation of SMERGE, MINOS 5.4 stopped after a too high number of errors during the evaluation of the objective function.

Problems	Viol	Viol/X	RG	RG/PI	Obj. value	Status
RAM200	1.8E-14	1.5E-17	1.5E-07	9.7E-08	9.3534284	Optimal
RAM400	7.3E-13	2.7E-17	2.7E-08	2.5E-08	9.3538004	Optimal
SMERGE	2.7E-10	6.0E-13	5.1E-03	3.5E-08	2,327,920	Optimal
MM-CH5						F
MM-CH9	1.9E-13	6.0E-18	7.8E-09	4.8E-09	5,249.1967	Optimal
MM-COL	2.2E-09	5.9E-14	2.6E-06	5.7E-07	4,205.799	Optimal
MM-SWE						F
MM-US						F

Table 12: MINOS 5.4 results for nonlinearly constrained convex formulations.

Problems	RG	RG/PI	Obj. value	Status
RAM200	2.54E-08	2.10E-08	9.3534506	Optimal
RAM400	3.85E-09	3.85E-09	9.3538135	Optimal
SMERGE				F
MM-CH5	3.17E-08	2.84E-08	2,802.6951	Optimal
MM-CH9	5.62E-10	3.62E-10	5,249.1967	Optimal
MM-COL	1.05E-05	2.43E-06	4,205.8	Optimal
MM-SWE	1.48E-07	1.01E-07	12,536.73	Optimal
MM-US				F

Table 13: MINOS 5.4 results for linearly constrained convex formulations.

8.5 MINOS 5.5

MINOS 5.5 claims that it found an optimal solution for the nonlinearly constrained convex formulation of SMERGE but it reports in the same time that 151 variables are nonoptimal. For the linearly constrained convex formulation of SMERGE, MINOS 5.5 stopped after a too high number of errors during the evaluation of the objective function. For the MM-US model, MINOS reports that it found the optimal solution, but in the same time it reports that some variables are not optimal or/and that feasibility is violated. Namely, for the nonlinearly constrained convex formulation, 2,250 variables are reported to be nonoptimal. For the linearly constrained convex formulation, 726 variables are reported to be nonoptimal and 2 to be infeasible, which is consistent with the high values of `FeasSum` and `FeasMax`. This could explain why the objective function values reported at optimality for the nonlinearly constrained convex formulation of MM-US, and, to a less extent, for the linearly constrained formulation, are pretty different from the values reported by other solvers.

Problems	`Feasible`	`Optimal`	`Obj. value`	`Status`
RAM200	2.1E-13	1.4E-06	9.35341	Optimal
RAM400	4.3E-06	1.0E-06	9.35345	Optimal
SMERGE	1.3E-08	2.3E-03	2,327,920	Optimal
MM-CH5	4.8E-06	9.9E-07	2,802.65	Optimal
MM-CH9	2.8E-13	9.9E-07	5,249.158	Optimal
MM-COL	4.4E-07	9.7E-07	4,201.585	Optimal
MM-SWE	8.8E-13	5.3E-06	12,535.9	Optimal
MM-US	6.0E-12	9.3E-07	7,837.447	Optimal

Table 14: MINOS 5.5 results for nonlinearly constrained convex formulations.

Problems	`FeasSum`	`FeasMax`	`Obj. value`	`Status`
RAM200	0.E+00	0.E+00	9.3533802	Optimal
RAM400	0.E+00	0.E+00	9.3536974	Optimal
SMERGE				F
MM-CH5	0.E+00	0.E+00	2,802.6834	Optimal
MM-CH9	1.2E-06	1.2E-06	5,249.1952	Optimal
MM-COL	0.E+00	0.E+00	4,201.8164	Optimal
MM-SWE	0.E+00	0.E+00	12,536.7841	Optimal
MM-US	6.0E-01	3.0E-01	7,859.4536	Optimal

Table 15: MINOS 5.5 results for linearly constrained convex formulations.

8.6 NLPHOPDM

No results are reported for NLPHOPDM in case of linearly constrained formulations since they are incompatible with the algorithm employed.

Problems	relPviol	relDviol	relGap	Obj. value	Status
RAM200	9.2E-08	1.5E-12	1.1E-11	9.353664	Optimal
RAM400	5.2E-13	3.8E-11	1.0E-06	9.354049	Optimal
SMERGE	8.2E-10	2.5E-08	8.5E-12	2,327,920.7	Optimal
MM-CH5	1.6E-07	6.4E-07	6.2E-10	2,802.695	Optimal
MM-CH9	8.8E-08	1.1E-07	8.5E-10	5,249.196	Optimal
MM-COL	9.0E-08	3.5E-07	2.6E-09	4,205.801	Optimal
MM-SWE	7.1E-07	1.4E-08	5.2E-11	12,535.77	Optimal
MM-US	5.3E-07	5.0E-07	5.9E-12	7,865.427	Optimal

Table 16: NLPHOPDM results for nonlinearly constrained convex formulations.

8.7 SNOPT

For the linearly constrained convex formulation of SMERGE, MINOS 5.5 stopped after a too high number of errors during the evaluation of the objective function. Note that sometimes SNOPT claims that it found the optimal solution but that some variables are not optimal. This occurs for Ramsey models and MM-US. For the linearly constrained convex formulations of RAM200 and RAM400, SNOPT reports that respectively 3 and 2 variables are not optimal. For the nonlinearly (resp. linearly) constrained convex formulation of MM-US, 2,485 (resp. 748) variables are reported to be nonoptimal. This could explain why the objective function values reported at optimality for the nonlinearly constrained convex formulation of MM-US, and, to a less extent, for the linearly constrained convex formulation, are pretty different from the values reported by other solvers.

Problems	Feasible	Optimal	Obj. value	Status
RAM200	1.1E-07	1.4E-06	9.35216	Optimal
RAM400	1.2E-07	1.3E-06	9.35135	Optimal
SMERGE	2.0E-12	1.1E-06	2,327,920.3	Optimal
MM-CH5	1.9E-08	1.6E-06	2,802.59	Optimal
MM-CH9	3.7E-07	1.2E-06	5,249.06	Optimal
MM-COL	4.9E-12	1.5E-06	4,197.28	Optimal
MM-SWE	7.6E-09	1.7E-06	12,536.7	Optimal
MM-US	3.5E-07	1.4E-06	7,843.95	Optimal

Table 17: SNOPT results for nonlinearly constrained convex formulations.

Problems	Optimal	Obj. value	Status
RAM200	1.6E-06	9.35249	Optimal
RAM400	1.3E-06	9.35239	Optimal
SMERGE			F
MM-CH5	1.2E-06	2,802.58	Optimal
MM-CH9	1.1E-06	5,249.05	Optimal
MM-COL	1.3E-06	4,199.20	Optimal
MM-SWE	1.6E-06	12,535.9	Optimal
MM-US	1.7E-06	7,861.74	Optimal

Table 18: SNOPT results for linearly constrained convex formulations.

8.8 LOQO

Next tables report results for LOQO 4.01, run on a 350 MHz machine, and LOQO 5.02, run on a 250 MHz machine.

After 2,602 iterations and 1,660 seconds, LOQO 4.01 reports that RAM400 is dual infeasible.

Problems	Pviol	Dviol	Pobj	Dobj	Status
RAM200	5.0E-07	5.9E-11	9.353664	9.353662	Optimal
RAM400					F
SMERGE	2.5E-08	3.2E-10	2,327,920.6	2,327,920.7	Optimal

Table 19: Detailed results for LOQO 4.01.

Note that both feasibility and optimality violations are well under the prespecified optimality tolerance chosen for NLPHOPDM. Indeed, for RAM200 (resp. SMERGE), LOQO 4.01 reports that the duality gap is equal to 2E-06 (resp. 0.107). By dividing this quantity by 9.35366 (resp. 2,327,920), this leads to a relative duality gap equal to 2E-07 (resp. 4.6E-08) which is lower than 1E-06, i.e. the prespecified optimality tolerance chosen for NLPHOPDM.

After 176 iterations and 24 seconds, LOQO 5.02 reports that it found a suboptimal solution for RAM400: `Pviol` is equal to 6.5E-07, `Dviol` to 1.3e-09, but `Pobj` to 9.35399 and `Dobj` to 9.35391.

Problems	Pviol	Dviol	Pobj	Dobj	Status
RAM200	7.5E-07	7.2E-11	9.353664	9.353662	Optimal
RAM400					F
SMERGE	2.5E-08	4.2E-10	2,327,920.6	2,327,920.7	Optimal

Table 20: Detailed results for LOQO 5.02.

References

[1] A. Altman and J. Gondzio, *Regularized symmetric indefinite systems in interior point methods for linear and quadratic optimization*, Optimization Methods and Software, 11-12 (2000), pp. 275–302.

[2] E. D. Andersen and Y. Y. Ye, *A computational study of the homogeneous algorithm for large-scale convex optimization*, Computational Optimization and Applications, 10 (1998), pp. 243–269.

[3] K. Anstreicher and J.-P. Vial, *On the convergence of an infeasible primal-dual IPM for convex programming*, Optimization Methods and Software, 3 (1994), pp. 273–283.

[4] P. Armand, J.-C. Gilbert, and S. Jegou-Jan, *A feasible BFGS interior-point algorithm for solving strongly convex optimization problems*, tech. report, INRIA, BP 105, 78153 Le Chesnay, France, 1998.

[5] O. Bahn, *Techniques de décomposition dans l'analyse des choix énergétiques et environnementaux*, PhD thesis, University of Geneva, 1996.

[6] O. Bahn, B. Büeler, S. Kypreos, and H.-J. Lüthi, *Modeling an international market of CO_2 emission permits*, International Journal of Global Energy Issues, 12 (1999), pp. 283–291.

[7] A. Brooke, D. Kendrick, and A. Meeraus, *GAMS: A User's Guide*, The Scientific Press, Redwood City, California, 1992.

[8] B. Büeler, *Computing Economic Equilibria and its Application to International Trade of CO_2 Permits: An Agents-Based Approach*, PhD thesis, Swiss Federal Institute of Technology, Zürich, December 1997.

[9] R. Byrd, M. E. Hribar, and J. Nocedal, *An interior point algorithm for large-scale nonlinear programming*, SIAM Journal on Optimization, 9 (1999), pp. 877–900.

[10] A. I. Cadena, *Models to assess the implications of the Kyoto protocol on the economy and energy system of Colombia*, PhD thesis, University of Geneva, May 2000 draft. Forthcoming.

[11] A. S. Drud, *CONOPT - A large-scale generalized reduced gradient code*, ORSA Journal on Computing, 6 (1994), pp. 207–216.

[12] ——, *Interaction between nonlinear programming and modeling systems*, Mathematical Programming, 79 (1997), pp. 99–124.

[13] A. V. Fiacco and G. P. McCormick, *Nonlinear Programming: Sequential Unconstrained Minimization Techniques*, John Wiley and Sons, New York, 1968.

[14] L. Fishbone and H. Abilock, *MARKAL: A linear programming model for energy systems analysis: technical description of the BNL version*, International Journal of Energy Research, 5 (1981), pp. 353–375.

[15] R. Fourer, D. Gay, and B. W. Kernighan, *AMPL: A Modeling Language for Mathematical Programming*, The Scientific Press, San Francisco, California, 1993.

[16] R. Gajulapalli and L. Lasdon, *Computational experience with a safeguarded barrier algorithm for sparse nonlinear programming*, tech. report, MSIS Department, University of Texas, Austin, December 1997.

[17] P. E. Gill, W. Murray, and M. A. Saunders, *SNOPT: A SQP algorithm for large-scale constrained optimization*, Tech. Report SOL 97-3, Stanford University, 1997.

[18] J. Gondzio, *HOPDM (version 2.12) – A fast LP solver based on a primal-dual interior point method*, European Journal of Operational Research, 85 (1995), pp. 221–225.

[19] ——, *Multiple centrality corrections in a primal-dual method for linear programming*, Computational Optimization and Applications, 6 (1996), pp. 137–156.

[20] B. Jansen, C. Roos, T. Terlaky, and J.-P. Vial, *Primal–dual target following algorithms for linear programming*, Annals of Operations Research, 62 (1996), pp. 197–231. K. Anstreicher and R. Freund (eds.).

[21] S. Kypreos, *Energy scenarios for Switzerland and emission control, estimated with a normative model*, tech. report, Paul Scherrer Institute, Switzerland, 1990.

[22] ——, *Assessment of CO_2 reduction policies for Switzerland*, International Journal of Global Energy Issues, 12 (1999), pp. 233–243.

[23] A. Manne, R. Mendelson, and R. Richels, *MERGE - A model for evaluating regional and global effects of GHG reduction policies*, Energy Policy, 23 (1995), pp. 17–34.

[24] A. Manne and C. Wene, *MARKAL-MACRO: A linked model for energy-economy analysis*, Tech. Report BNL-47161, Brookhaven National Laboratory, NJ, 1992.

[25] S. Mehrotra, *On the implementation of a primal–dual interior point method*, SIAM Journal on Optimization, 2 (1992), pp. 575–601.

[26] B. Murtagh and M. Saunders, *MINOS 5.5 User's Guide*, Department of Operations Research, Stanford University, 1983. Revised 1987, 1993, 1995, Jul. 1998.

[27] I. Nyström and C.-O. Wene, *Energy-economy linking in MARKAL-MACRO: interplay of nuclear conservation and CO_2 policies in Sweden*, International Journal of Environment and Pollution, (1998). Accepted for publication.

[28] F. Ramsey, *A mathematical theory of saving*, Economic Journal, (1928), pp. 543–559.

[29] P. Tseng, *Private communication.* Department of Energy, USA, 1997.

[30] R. J. Vanderbei and D. Shanno, *Interior-point methods for nonconvex nonlinear programming: orderings and higher-order methods*, Mathematical Programming, 87 (2000), pp. 303–316.

[31] J.-P. Vial, *Computational experience with a primal-dual IPM for smooth convex programming*, Optimization Methods and Software, 3 (1994), pp. 285–316.