Available online at www.sciencedirect.com
ScienceDirect
IFAC PapersOnLine 51-5 (2018) 120–125

MPC-DICE: An open-source Matlab implementation of receding horizon solutions to DICE ⋆

Timm Faulwasser * **Christopher M. Kellett** ** **Steven R. Weller** **

* *Institute for Automation and Applied Informatics, Karlsruhe Institute of Technology, 76344 Eggenstein-Leopoldshafen, Germany*
** *School of Electrical Engineering and Computing, University of Newcastle, Callaghan, New South Wales 2308, Australia*

Abstract: The most widely used integrated assessment model for studying the economics of climate change is the Dynamic Integrated model of Climate and Economy (DICE). DICE is a nonlinear, time-varying discrete-time system whereby important quantities are calculated by solving an associated optimal control problem. Here, we describe an open-source Matlab implementation of the DICE model which uses Model Predictive Control (MPC) to approximately solve the DICE optimal control problem.

Keywords: Economics of Climate Change, Climate Models, Economic Models, Computer Software

1. INTRODUCTION

Studies in the economics of climate change have been dominated by three independent integrated assessment models: DICE (Dynamic Integrated model of Climate and Economy) (Nordhaus, 1992), PAGE (Policy Analysis of the Greenhouse Effect) (Hope, 2015), and FUND (Climate Framework for Uncertainty, Negotiation, and Distribution) (Anthoff and Tol, 2013). In part, the importance of these three models stems from their use by policy makers and stakeholders to derive estimates of the *social cost of carbon of carbon dioxide* ($SC\text{-}CO_2$ – sometimes also called the social cost of carbon), see (Interagency Working Group on Social Cost of Carbon, U.S. Government, 2013)

Moreover, a recent extensive study (National Academies of Sciences, Engineering, and Medicine, 2017) discusses the need of revising the models used in estimating the $SC\text{-}CO_2$. While this report contains many detailed recommendations, it identified three overarching themes required of future integrated assessment models: (i) *incorporation of uncertainty*, (ii)*modularity*, and (iii) *transparency*.

In this context, the DICE model provides a useful starting point. Indeed, an open-source implementation of the model (in the GAMS language) has long been available online with a user manual (Nordhaus, 2013, 2016; Nordhaus and Sztorc, 2013). Furthermore, the DICE model is modular in nature, consisting of clearly defined climate, carbon, and socioeconomic submodels. Incorporating uncertainty into the DICE model is the subject of ongoing work.

The DICE model was originally proposed in (Nordhaus, 1992), with updates in (Nordhaus, 2008), (Nordhaus, 2014), and (Nordhaus, 2017). The model consists of a nonlinear, time-varying discrete-time dynamical system with two inputs. The inputs are usually calculated by solving a finite-horizon discounted Optimal Control Problem (OCP); i.e., an optimal control problem involving a discounted cost function.

While the available (GAMS) implementation (Nordhaus, 2013) specifies a finite-horizon OCP, conceptually an infinite-horizon OCP is easier to justify. In particular, depending on the parameters used in the model, important system behavior may not be observed within the specified finite-horizon. However, solving the infinite-horizon OCP analytically appears to be intractable.

Similar economic problems were solved in (Grüne et al., 2015) using Model Predictive Control (MPC) (also called receding horizon control) primarily as an approximation technique for the underlying infinite-horizon OCP. Furthermore, in (Grüne, 2016) it was demonstrated that, under certain assumptions, solution trajectories constructed using MPC provide quantifiable approximations to the solution of the infinite-horizon OCP.

An MPC implementation of the DICE model was first proposed in (Weller et al., 2015) and subsequently used in (Hafeez et al., 2017). However, the MPC implementation of (Weller et al., 2015) and (Hafeez et al., 2017) made use of the available GAMS code (Nordhaus, 2013) to solve each finite-horizon OCP and then used a Matlab wrapper to advance time and piece together the solution trajectories.

Recently, we provided a native Matlab implementation of the DICE model with a finite-horizon OCP (Faulwasser et al., 2016). The work reported here extends the work of (Faulwasser et al., 2016) with an MPC implementation fully in Matlab, which we term *MPC-DICE*.

⋆ TF acknowledges support from the Daimler Benz Foundation. The authors are supported by the Australian Research Council under ARC-DP180103026.

2405-8963
Peer review under responsibility of International Federation of Automatic Control.
10.1016/j.ifacol.2018.06.221

The paper is organized as follows. Section 2 provides a detailed description of the DICE model and associated OCP. (Default model parameters are contained in an Appendix.) Section 3 describes how to reformulate the DICE model to facilitate the MPC implementation and, in particular, the computation of the social cost of carbon. Section 4 provides details on the software requirements for using MPC-DICE, describes the various files making up MPC-DICE, and highlights some of the important options. As we make some minor modifications to the original DICE model, some comparative results are provided in Section 5. The MPC-DICE code is available for download at (Faulwasser et al., 2018).

2. MODEL AND OPTIMAL CONTROL PROBLEM

The MPC-DICE model operates on five year time steps beginning from 2015. To formalize this, let $t_0 = 2015$, $\Delta = 5$, and $i = 1, 2, 3, \ldots$ be the discrete time index.[1] Then

$$t = t_0 + \Delta \cdot (i-1) \tag{1}$$

yields $t = 2015, 2020, 2025 \ldots$ as desired.

The MPC-DICE model has eleven state variables: two variables to model the global climate in the form of atmospheric and oceanic temperature anomalies (T_{AT} and T_{LO}, respectively, in units of °C); three variables to model the global carbon cycle in the form of carbon concentrations in the atmosphere, upper ocean, and lower ocean (M_{AT}, M_{UP}, and M_{LO}, respectively, in units of GtC); and one state for global capital (K, in units of trillions 2010USD). Additional state variables describe the global population (L in millions of people), total factor productivity (A), the emissions intensity of economic activity (σ in units of $\mathrm{GtCO_2}$ per trillions of 2010USD), emissions due to land use changes (E_{Land} in units of $\mathrm{GtCO_2}$), and radiative forcings due to non-CO_2 greenhouse gases (F_{EX} in units Wm^{-2}). Decision variables or control inputs are the dimensionless emissions mitigation rate (μ) and the dimensionless savings rate (s) where the latter is the ratio of investment to net economic output.

To facilitate a compact representation, define the state vectors

$$T \doteq [T_{\mathrm{AT}}\ T_{\mathrm{LO}}]^\top, \quad M \doteq [M_{\mathrm{AT}}\ M_{\mathrm{UP}}\ M_{\mathrm{LO}}]^\top,$$

and the matrices

$$\Phi_T \doteq \begin{bmatrix} \phi_{11} & \phi_{12} \\ \phi_{21} & \phi_{22} \end{bmatrix}, \qquad B_T \doteq \begin{bmatrix} \xi_1 \\ 0 \end{bmatrix},$$

$$\Phi_M \doteq \begin{bmatrix} \zeta_{11} & \zeta_{12} & 0 \\ \zeta_{21} & \zeta_{22} & \zeta_{23} \\ 0 & \zeta_{32} & \zeta_{33} \end{bmatrix}, \qquad B_M \doteq \begin{bmatrix} \xi_2 \\ 0 \\ 0 \end{bmatrix}.$$

We will also use two intermediate quantities given by[2]

$$\theta_1(i) \doteq \frac{p_b}{1000 \cdot \theta_2}(1-\delta_{pb})^{i-1} \cdot \sigma(i), \tag{3}$$

$$Y(i) \doteq A(i) K(i)^\gamma \left(\tfrac{L(i)}{1000}\right)^{1-\gamma}. \tag{4}$$

The full DICE dynamics are given by:

$$\begin{aligned} T(i+1) &= \Phi_T T(i) \\ &\quad + B_T\left(\eta \log_2\left(\frac{M_{\mathrm{AT}}(i)}{M_{\mathrm{AT},1750}}\right) + F_{\mathrm{EX}}(i)\right) && (5)\\ M(i+1) &= \Phi_M M(i) + B_M E(i) && (6)\\ K(i+1) &= (1-\delta_K)^\Delta K(i) \\ &\quad + \Delta\left(\frac{1-\theta_1(i)\mu(i)^{\theta_2}}{1+a_2\, T_{\mathrm{AT}}(i)^{a_3}}\right) Y(i) s(i), && (7)\\ \sigma(i+1) &= \sigma(i)\exp\left(-g_\sigma(1-\delta_\sigma)^{\Delta(i-1)}\Delta\right), && (8)\\ L(i+1) &= L(i)\left(\frac{L_a}{L(i)}\right)^{\ell_g}, && (9)\\ A(i+1) &= \frac{A(i)}{1-g_A\exp(-\delta_A\Delta(i-1))}, && (10)\\ E_{\mathrm{Land}}(i+1) &= E_{L0}\cdot(1-\delta_{EL})^i && (11)\\ F_{\mathrm{EX}}(i+1) &= f_0 + \min\left\{\frac{(f_1-f_0)i}{t_f},\ f_1-f_0\right\} && (12) \end{aligned}$$

where emissions (E in units of $\mathrm{GtCO_2}$), driving (6), are also an intermediate quantity given by

$$E(i) = \Delta\left(\sigma(i)(1-\mu(i))Y(i) + E_{\mathrm{Land}}(i)\right). \tag{13}$$

Consumption (C) can be viewed as an output of the model and is given by

$$C(i) = \Delta\left(\frac{1-\theta_1(i)\mu(i)^{\theta_2}}{1+a_2\, T_{\mathrm{AT}}(i)^{a_3}}\right) Y(i)(1-s(i)). \tag{14}$$

Note that, in the economics literature, (5)–(7) are usually referred to as "endogenous variables", while (9)–(12) are usually referred to as "exogenous variables". This is due to the fact that each of the latter set of states is independent of any other state.

Finally, for the purposes of defining an optimization problem, utility is given by

$$U(C(i), L(i)) = L(i)\left(\frac{\left(\frac{1000C(i)}{L(i)}\right)^{1-\alpha} - 1}{1-\alpha}\right). \tag{15}$$

Optimal pathways are derived by maximizing the social welfare; i.e., by solving the following OCP

$$\max_{\mathbf{s},\mu}\ \Delta \cdot scale1 \cdot \sum_{i=1}^{\infty} \frac{U(C(i),L(i))}{(1+\rho)^{\Delta(i-1)}} - scale2 \tag{16a}$$

$$\begin{aligned} \text{subject to} \quad & (5-12) \\ & \mu(1) = \mu_0 \\ & 0 \le \mu(i) \le 1,\ \forall i \\ & 0 \le s(i) \le 1,\ \forall i \end{aligned} \tag{16b}$$

[1] Note that t_0 is coded as a parameter so that it is straightforward to replicate results from DICE2013 where $t_0 = 2010$.

[2] In (Nordhaus, 2013, 2016), θ_1 is called `cost1`. Furthermore, as the implementation of (7) in (Nordhaus, 2013, 2016) contains minor errors, we here explicitly mention nomenclature used in (Nordhaus, 2013) and the economics literature. The *damages fraction* is given by

$$\mathtt{DAMFRAC}(i) = \frac{a_2 T_{\mathrm{AT}}(i)^{a_3}}{1+a_2 T_{\mathrm{AT}}(i)^{a_3}}$$

and the damages component of (7) (i.e., the fraction of output remaining after climate damages) is then written as $(1-\mathtt{DAMFRAC}(i))$. Similarly, the *mitigation* or *abatement fraction* is given by

$$\mathtt{ABATEFRAC}(i) = \theta_1(i)\mu(i)^{\theta_2} \tag{2}$$

and the cost of abatement in (7) (i.e., the fraction of output remaining after spending on abatement) is written as $(1-\mathtt{ABATEFRAC}(i))$.

Note that the constraint $\mu(1) = \mu_0$ is imposed since $\mu(1)$ corresponds to the mitigation rate in the year t_0; something we can reasonably estimate.

The SC-CO_2 is given by the ratio of the marginal welfare with respect to emissions and with respect to consumption (Newbold et al., 2013):

$$\mathrm{SCC}(i) = -1000 \cdot \frac{\partial W/\partial E(i)}{\partial W/\partial C(i)}. \tag{17}$$

3. RECEDING HORIZON SPECIFIC REFORMULATION OF THE OCP

To solve the DICE OCP in a receding horizon manner, we use an equivalent reformulation. We begin by defining the augmented state vector Note that the augmented vector $\tilde{x}(i) \in \mathbb{R}^{12}$ collects the state variables of (5–12) plus the time index i. As we show later including the time as a state variable will allow rewriting the dynamics underlying the original DICE OCP (16) with a time-invariant state transition map. The vector $x_{aux}(i) \in \mathbb{R}^5$ collects the emissions (13), consumption (14), inputs $\mu(i)$ and $s(i)$ at time i, and the extra state

$$J(i) = \sum_{j=1}^{i} \frac{U(C(j), L(j))}{(1+\rho)^{\Delta(j-1)}},$$

which is used to define the objective (social welfare).

Moreover, using

$$x(i) \doteq \left[\tilde{x}(i)^\top \; x_{aux}(i)^\top\right]^\top$$

and the shifted input variables

$$w(i) \doteq \left[\mu(i+1) \; s(i+1)\right]^\top,$$

we can rewrite the dynamics underlying the original DICE OCP (16) as follows:

$$x(i+1) = f(x(i), w(i)), \quad x(1) = v. \tag{18}$$

The first component of the righthand-side function $f : \mathbb{R}^{17} \times \mathbb{R}^2 \to \mathbb{R}^{17}, f = [f_1, \dots, f_{17}]^\top$ is given by

$$f_1(x, w) \doteq x_1 + 1,$$

and the components $k = 2, \dots, 12$ are given by (5–12). For $k = 13$ we obtain from (13) and (4)

$$\begin{aligned} E(i+1) &= f_{13}(x(i), w(i)) \\ &= \Delta\Big(\sigma(i+1)(1-\mu(i+1))Y(i+1) + E_{\mathrm{Land}}(i+1)\Big) \\ &= \Delta\Big(f_8(x(i), w(i)) \cdot (1 - w_1(i)) \cdot f_{10}(x(i), w(i)) \\ &\quad \cdot f_7(x(i), w(i))^\gamma \cdot \left(\tfrac{f_9(x(i), w(i))}{1000}\right)^{1-\gamma} \\ &\quad + E_{L0} \cdot (1 - \delta_{EL})^i\Big). \end{aligned} \tag{19}$$

In other words, we can rewrite the emissions explicitly as a state using (7)–(10) to expand f_7, f_8, f_9, and f_{10}. Immediately from the above, we obtain the initial emissions

$$E(1) = \Delta\left(x_8(1)(1 - x_{15}(1))x_{10}(1)x_8(1)^\gamma \left(\tfrac{x_9(1)}{1000}\right)^{1-\gamma} + E_{L0}\right). \tag{20}$$

Similarly, we may rewrite the consumption as a state equation as

$$\begin{aligned} C(i+1) &= f_{14}(x(i), w(i)) \\ &= \Delta\left(\frac{1 - \theta_1(i+1)\mu(i+1)^{\theta_2}}{1 + a_2\, T_{\mathrm{AT}}(i+1)^{a_3}}\right) Y(i+1)(1 - s(i+1)) \\ &= \Delta\left(\frac{1 - \theta_1(i+1)w_1(i)^{\theta_2}}{1 + a_2\, f_2(x(i), w(i))^{a_3}}\right) \cdot f_{10}(x(i), w(i)) \\ &\quad \cdot f_7(x(i), w(i))^\gamma \cdot \left(\frac{f_9(x(i), w(i))}{1000}\right)^{1-\gamma} \\ &\quad \cdot (1 - w_2(i)) \end{aligned} \tag{21}$$

with initial condition given by

$$C(1) = \Delta\left(\frac{1 - \theta_1(1)x_{15}(1)^{\theta_2}}{1 + a_2\, x_2(1)^{a_3}}\right) \cdot x_{10}(1) \cdot x_8(1)^\gamma \cdot \left(\tfrac{x_8(1)}{1000}\right)^{1-\gamma} \cdot (1 - x_{16}(1)). \tag{22}$$

The final three states are given by

$$x_{15}(i+1) = w_1(i), \quad x_{15}(1) = v_{15} \tag{23}$$

$$x_{16}(i+1) = w_2(i), \quad x_{16}(1) = v_{16} \tag{24}$$

$$x_{17}(i+1) = x_{17}(i) + \frac{U(x_{12}(i), x_9(i))}{(1+\rho)^{\Delta(i-1)}}, \quad x_{17}(1) = 0. \tag{25}$$

Observe that the initial condition $x_{14}(1) = C(1)$ depends on the (unshifted) inputs at time $i = 1$; i.e. it depends on $\mu(1) = x_{15}(1)$ and $s(1) = x_{16}(1)$. Likewise the initial condition $x_{13}(1) = E(1)$ depends on $\mu(1) = x_{15}(1)$.

To handle this dependence in the optimization, we introduce the auxiliary decision variable $v \in \mathbb{R}^{17}$ and the additional constraint

$$x(1) = v.$$

Now, we can summarize the equivalent (finite-horizon) reformulation of the DICE-OCP (16) based on the augmented (18) as follows

$$\max_{\mathbf{w},\, v} \; x_{17}(N) \tag{26a}$$

subject to

$$x(j+1) = f(x(j), w(j)), \quad j = 1, \dots, N-1 \tag{26b}$$

$$x(1) = v \tag{26c}$$

$$v_k = x_k(1),\; k \in \{1, \dots, 17\} \setminus \{15, 16\} \tag{26d}$$

$$v_k \in [0,\, 1], \quad k = 15, 16 \tag{26e}$$

$$w(j) \in [0,\, 1] \times [0, 1], \quad j = 1, \dots, N. \tag{26f}$$

In order to obtain a receding horizon variant of the original DICE-OCP, we define a second optimization problem as follows:

$$\max_{\mathbf{w}} \; x_{17}(N) \tag{27a}$$

subject to

$$x(j+1) = f(x(j), w(j)), \quad j = i, \dots, N-1 \tag{27b}$$

$$x(1) = x(i) \tag{27c}$$

$$w(j) \in [0,\, 1] \times [0, 1], \quad j = 1, \dots, N, \tag{27d}$$

which differs from OCP (26) in that the initial condition $x(1)$ is available from the previous optimization via the variable $x(i)$. Consequently, the extra decision variable v is not required to capture the dependence of $x_{13}(1) = E(1)$ and $x_{14}(1) = C(1)$ on the decision variables at the same time; i.e., $\mu(1)$ and $s(1)$. Solving either OCP (26) or OCP (27), we obtain the following data:

- The optimal state trajectory $x^\star(j), j = 1, \ldots, N$, which contains the savings rate and the mitigation rate as
$$\mu^\star(j) = x^\star_{15}(j) \quad \text{and} \quad s^\star(j) = x^\star_{16}(j).$$
- The optimal adjoint variables $\lambda^\star_C(j)$ and $\lambda^\star_E(j)$ which are given by the Lagrange multipliers associated to the equality constraints implied by the dynamics of $E(j) = x_{13}(j)$ and $C(j) = x_{14}(j)$.

Hence, the SC-CO_2 at time j is obtained by
$$\text{SCC}(j) = -1000 \cdot \frac{\partial W/\partial E(j)}{\partial W/\partial C(j)} = -1000 \cdot \frac{\lambda^\star_E(j)}{\lambda^\star_C(j)}.$$

Now, we can state the receding horizon approximation of the DICE-OCP (16) as follows:

Algorithm 1 MPC-DICE

1: Define computation horizon N_{sim} and prediction horizon N,
2: **if** $i = 1$ **then**
3: Solve OCP (26)
4: Set $x(1) = x^\star(1), \lambda_E(1) = \lambda^\star_E(1), \lambda_C(1) = \lambda^\star_C(1)$.
5: **for** $i = 2, \ldots, N_{sim}$ **do**
6: Solve OCP (27) for the initial condition $x_1 = x(i-1)$.
7: Set $x(i) = x^\star(1), \lambda_E(i) = \lambda^\star_E(1), \lambda_C(i) = \lambda^\star_C(1)$.
8: Return $x(j)$, $\lambda_E(j)$ and $\lambda_C(j), j = 0, \ldots, N_{sim}$.

4. DESCRIPTION OF CODE

The implementation of MPC-DICE makes use of the CasADi framework for algorithmic differentiation and numerical optimization (Andersson and Gillis, 2016) in conjunction with Matlab. The resulting nonlinear programs are solved via IPOPT (Wächter and Biegler, 2005) for which an executable is provided with CasADi. Version 3.2.1 of CasADi is used and, hence, Matlab 2014a or later is generally required. Appropriate binaries[3] of CasADi v.3.2.1 are available at (Andersson and Gillis, 2016).

Similar to CasADi, MPC-DICE is distributed under the GNU Lesser General Public License (LGPL), and hence the code can be used royalty-free even in commercial applications.

The implementation of MPC-DICE is organized in five files with an additional test file provided. These files are as follows:

- `MPCDICE_mc.m` is the top-level file and calls the subsequent five files. This file executes Algorithm 1 and also includes code allowing the solution of a single-shot finite-horizon optimal control problem.
- `assign_parameters.m` is a function file that returns the parameters of the MPC-DICE model and optimal control problem. The code available at (Faulwasser et al., 2018) provides three such files:
  · `assign_parameters_v2013.m` and
  · `assign_parameters_v2016.m`
  contain the parameter sets for DICE2013R and DICE2016R, respectively. We do not recommend editing these parameters since, in some sense, they provide a base case for comparison. Hence, we also explicitly provide `assign_parameters_template.m`.
- `Construct_NLP.m` is a function that constructs symbolic CasADi formulations of OCP (26) and OCP (27). To speed-up the receding horizon computations, OCP (27) considers the initial condition $x(1) = x_1$ as a free parameter which is passed during run-time to the corresponding function call.
- `dice_dynamics.m` implements the discrete-time dynamical system given by (18). While not directly called by `MPCDICE_mc.m`, this function is called by `Construct_NLP.m`.
- `set_initial_conditions.m` calculates and returns the initial conditions for all states of (18) from data given in `assign_parameters.m`. This function is called by `MPCDICE_mc.m` and by `Construct_NLP.m`. Importantly, in the former call the initial conditions are returned as standard Matlab variables, whereas in the latter call the initial conditions are returned as CasADi variables.
- `MPCDICE_test.m` is a function used to test the CasADi installation.

There are two particular types of variation likely to be of interest: parametric changes and structural changes. Changes to parameters are preferably limited to the file `assign_parameters.m`. Structural changes require editing both `dice_dynamics.m` and `set_initial_conditions.m`. This latter requirement stems from the fact that a subset of the initial conditions of (18) are calculated based on the equations of the system dynamics.

4.1 Single OCP versus Receding Horizon OCP

In the top-level file `MPCDICE_mc.m`, it is possible to switch between different operation modes of MPC-DICE. The important section of this file (starting right after the initial comment and some workspace housekeeping) is listed below. Due to space limitations, only selected code comments are printed.

```
...
%%=========================================================
% Define data of the DICE MPC-loop & construct NLP
%==========================================================

% Set parameters using an appropriate function
Params = assign_parameters_v2016;

Data.N = Params.N;   % Prediction horizon
Data.step = 1;       % 1 == 1-step MPC strategy
Data.t0 = 1;         % initial simulation time; >=1
Data.tf = 10;        % final simulation time
Data.nx = 1+6+9+1;   % total # of states: time;
                     %   6 "endogenous" states;
                     %   9 auxilliary states including
                     %   5 "exogenous" states,
                     %   consumption, emissions, and
                     %   shifted inputs; objective
Data.nu = 2;         % # of inputs

Params.x0 = set_initial_conditions(Data.t0,Params);
...
```

MPC-DICE allows the following computations:

[3] After downloading an appropriate binary, be sure to add CasADi to your Matlab path as described at (Andersson and Gillis, 2016).

- Solving a single DICE OCP.
- Solving the DICE problem in a receding horizon fashion to approximate the infinite horizon solution.

With `Params.N = 30` in the "assign parameters" function, using the following setting

```
...
Data.step = 1; % 1 == 1-step MPC strategy
Data.t0 = 1;   % initial simulation time; >=1
Data.tf = 1;   % final simulation time
...
```

will solve a single DICE OCP over a fixed horizon of $N = 30$ time steps (which corresponds to a time horizon of 150 years with the default $\Delta = 5$).

Solving the DICE problem in a receding horizon fashion to approximate the infinite horizon solution is accomplished with

```
...
Data.step = 1; % 1 == 1-step MPC strategy
Data.t0 = 1;   % initial simulation time; >=1
Data.tf = 60;  % final simulation time
...
```

This considers an overall horizon of $N_{sim} = \mathtt{tf} = 60$ and will solve a sequence of 60 DICE OCPs in receding horizon fashion, whereby each OCP considers an optimization horizon of $N = 30$ time steps and from each of these OCPs only the first step of the control action is considered for the output data.

5. COMPARATIVE RESULTS

As previously mentioned, the GAMS implementation of the DICE model in (Nordhaus, 2013, 2016) (and hence, deliberately, the code of Faulwasser et al. (2016)) contains minor errors. Therefore, we here present results using the default values for DICE 2016R and MPC-DICE.

In particular, (Nordhaus, 2016) computes the damages component of (7) as

$$1 - a_2 T_{\mathrm{AT}}(i)^{a_3} - \theta_1(i)\mu(i)^{\theta_2},$$

which, depending on the parameters chosen, can lead to an inappropriate sign change. Additionally, in (Nordhaus, 2016), the radiative forcing term in (5) is dependent on the mass of atmospheric carbon at the *current* time step rather than the *previous* time step; i.e., (5) is dependent on $M_{\mathrm{AT}}(i+1)$.

Correcting these two inconsistencies does not lead to significant quantitative differences in important outputs when using the default parameters. In Table 1 we show the peak warming (i.e., highest value attained of the atmospheric temperature) and the social cost of carbon values for the years 2015, 2025, and 2050 as calculated using the single-shot DICE 2016R OCP Faulwasser et al. (2016) and the receding horizon version of MPC-DICE Faulwasser et al. (2018).

	DICE 2016R	MPC-DICE
Peak Warming	4.15 °C	4.21 °C
SC-CO_2(2015)	US$ 30.75	US$ 29.26
SC-CO_2(2025)	US$ 43.62	US$ 41.34
SC-CO_2(2050)	US$ 91.32	US$ 85.47

Table 1. Selection of key outputs from DICE 2016R and MPC-DICE. Note that currency is 2010 US dollars.

6. CONCLUSIONS AND FUTURE WORK

In this paper we described an open-source Matlab implementation of the DICE IAM with its associated optimal control problem solved in a receding horizon fashion. Importantly, this implementation corrects two long-standing errors in previously available DICE implementations. While these errors do not have a significant impact on the predictions of the model using the the nominal parameters, alternate parameter sets can result in large differences. Additionally, the MPC implementation allows approximate solution of the infinite-horizon optimal control problem, avoiding the turnpike-like effects that result from solving a finite-horizon optimal control problem. Finally, with the code publicly and freely available and natively in Matlab, will make the DICE IAM more accessible to the systems and control community.

APPENDIX A: DEFAULT INITIAL CONDITIONS

	$T_{\mathrm{AT}}(0)$	$T_{\mathrm{LO}}(0)$	$K(0)$
2013R	0.8	0.0068	135
2016R	0.85	0.0068	223

	$M_{\mathrm{AT}}(0)$	$M_{\mathrm{UP}}(0)$	$M_{\mathrm{LO}}(0)$
2013R	830.4	1527	10010
2016R	851	460	1740

The parameters for calculating $\sigma_0 = \frac{e_0}{q_0(1-\mu_0)}$:

	e_0	q_0	μ_0
2013R	33.61	63.69	0.039
2016R	35.85	105.5	0.03

APPENDIX B: DEFAULT PARAMETER VALUES

In the table below, a blank entry indicates that the parameter was not changed in the 2016R version of the model.

Parameter	Value DICE2013R	Value DICE2016R
Δ	5	
t_0	2010	2015
N	60	100
μ_0	0.039	0.03
Climate diffusion parameters		
ϕ_{11}	0.8630	0.8718
ϕ_{12}	0.0086	0.0088
ϕ_{21}	0.025	0.025
ϕ_{22}	0.975	0.975
Carbon cycle diffusion parameters		
ζ_{11}	0.912	0.88
ζ_{12}	0.03833	0.196
ζ_{21}	0.088	0.12
ζ_{22}	0.9592	0.797
ζ_{23}	0.0003375	0.001465
ζ_{32}	0.00250	0.007
ζ_{33}	0.9996625	0.99853488
Other geophysical parameters		
η	3.8	3.6813
ξ_1	0.098	0.1005
ξ_2	3/11	
$M_{\mathrm{AT},1750}$	588	
E_{L0}	3.3	2.6
δ_{EL}	0.2	0.115
f_0	0.25	0.5
f_1	0.70	1.0
t_f	18	17
Socioeconomic parameters		
γ	0.3	
θ_2	2.8	2.6
a_2	0.00267	0.00236
a_3	2	
δ_K	0.1	
α	1.45	
ρ	0.015	
L_0	6838	7403
L_a	10500	11500
ℓ_g	0.134	
A_0	3.80	5.115
g_a	0.079	0.076
δ_A	0.006	0.005
p_b	344	550
δ_{pb}	0.025	
σ_0	0.5491	0.3503
g_σ	0.01	0.0152
δ_σ	0.001	
$scale1$	0.016408662	0.030245527
$scale2$	3855.106895	10993.704

REFERENCES

Andersson, J. and Gillis, J. (2016). CasADi 3.0.0. Available at `https://github.com/casadi/casadi/wiki`.

Anthoff, D. and Tol, R.S. (2013). The uncertainty about the social cost of carbon: A decomposition analysis using FUND. *Climatic Change*, 117(3), 515–530.

Faulwasser, T., Kellett, C.M., and Weller, S.R. (2016). DICE2013R-mc: Dynamic Integrated model of Climate and Economy 2013R - Matlab and CasADi. Available at `https://github.com/cmkellett/DICE2013R-mc`.

Faulwasser, T., Kellett, C.M., and Weller, S.R. (2018). MPC-DICE: Model Predictive Control – Dynamic Integrated model of Climate and Economy. Available at `https://github.com/cmkellett/MPC-DICE`.

Grüne, L. (2016). Approximation properties of receding horizon optimal control. *Jahresber Dtsch Math-Ver*, 118, 3–37.

Grüne, L., Semmler, W., and Stieler, M. (2015). Using nonlinear model predictive control for dynamic decision problems in economics. *J. Econ. Dyn. Control*, 60, 112–133.

Hafeez, S., Weller, S.R., and Kellett, C.M. (2017). Impact of climate model parametric uncertainty in an MPC implemenation of the DICE integrated assessment model. In *Proceedings of the 2017 IFAC World Congress*. Toulouse, France.

Hope, C. (2015). The $10 trillion value of better information about the transient climate response. *Phil. Trans. R. Soc. A*, 373(2054), 20140429.

Interagency Working Group on Social Cost of Carbon, U.S. Government (2013). Technical Update of the Social Cost of Carbon for Regulatory Impact Analysis - Under Executive Order 12866. URL `http://bit.ly/1JBGowf`.

National Academies of Sciences, Engineering, and Medicine (2017). *Valuing Climate Damages: Updating Estimation of the Social Cost of Carbon Dioxide*. The National Academies Press. doi:10.17226/24651.

Newbold, S., Griffiths, C., Moore, C., Wolverton, A., and Kopits, E. (2013). A rapid assessment model for understanding the social cost of carbon. *Climate Change Economics*, 4(01), 1350001.

Nordhaus, W. (2014). Estimates of the social cost of carbon: Concepts and results from the DICE-2013R model and alternative approaches. *J. Assoc. Environ. Resour. Econ.*, 1(1/2), 273–312.

Nordhaus, W.D. (2008). *A Question of Balance: Weighing the Options on Global Warming Policies*. Yale University Press.

Nordhaus, W.D. (2013). URL `http://www.econ.yale.edu/~nordhaus/homepage/Web-DICE-2013-April.htm`.

Nordhaus, W.D. (2016). DICE 2016R GAMS Code. URL `http://bit.ly/2lFKyMd`.

Nordhaus, W.D. (2017). Revisiting the social cost of carbon. *Proc. Natl. Acad. Scie. USA (PNAS)*, 114(7), 1518–1523.

Nordhaus, W.D. and Sztorc, P. (2013). *DICE 2013R: Introduction and User's Manual*, second edition. URL `http://www.econ.yale.edu/~nordhaus/homepage/`.

Nordhaus, W.D. (1992). An optimal transition path for controlling greenhouse gases. *Science*, 258, 1315–1319.

Wächter, A. and Biegler, L.T. (2005). On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming. *Mathematical Programming*, 106(1), 25–57. doi:10.1007/s10107-004-0559-y.

Weller, S.R., Hafeez, S., and Kellett, C.M. (2015). A receding horizon control approach to estimating the social cost of carbon in the presence of emissions and temperature uncertainty. In *Proc. 54th IEEE Conf. Decis. Control (CDC)*. Osaka, Japan.