



IonMonger: a free and fast planar perovskite solar cell simulator with coupled ion vacancy and charge carrier dynamics

N. E. Courtier¹ · J. M. Cave² · A. B. Walker² · G. Richardson¹ · J. M. Foster³

Published online: 13 September 2019
© The Author(s) 2019

Abstract

Details of an open-source planar perovskite solar cell simulator, which includes ion vacancy migration within the perovskite layer coupled to charge carrier transport throughout the perovskite and adjoining transport layers in one dimension, are presented. The model equations are discretised in space using a finite element scheme, and temporal integration of the resulting system of differential algebraic equations is carried out in MATLAB. The user is free to modify device parameters, as well as the incident illumination and applied voltage. Time-varying voltage and/or illumination protocols can be specified, e.g. to simulate current–voltage sweeps, or to track the open-circuit conditions as the illumination is varied. Typical simulations, e.g. current–voltage sweeps, only require computation times of seconds to minutes on a modern personal computer. An example set of hysteretic current–voltage curves is presented.

Keywords Perovskite solar cell · Drift–diffusion · Device simulation · Ion vacancy migration

1 Introduction

Perovskite solar cells (PSCs) are a promising thin-film technology that, due to their rapid rise in power conversion efficiency to 22.7% [15], are attracting a lot of interest and research effort in the photovoltaic community. However, PSCs display unusual transient behaviour (exemplified by current–voltage hysteresis) in the order of seconds to days which affects the power output of the device [24]. The consensus in the literature is that this slow (compared to the timescale of electronic motion) behaviour is due to the motion of mobile ion vacancies within the perovskite layer. The species of ion vacancy deemed most likely to be responsible for the behaviour observed in the order of seconds to

minutes is that of the halide (e.g. iodide, I^-) ion vacancy due to its high mean density and high diffusion coefficient (compared to the other ionic species) predicted by DFT calculations [8, 26]. Visual evidence of iodide ion migration within a perovskite film has also been obtained experimentally [7]. Recent reviews of the outstanding challenges in the field of perovskite solar cells have been given by Snaith [23], Egger et al. [9] and Phung and Abate [17].

Due to the existence of mobile ion vacancies, the perovskite layer must be treated as a mixed ionic–electronic conductor for the purpose of device simulation. The first works [2, 11, 25] to apply numerical methods to PSC modelling reported that their simulations suffered from prohibitively long calculation times and inaccuracies in solution for realistic values of the parameters. A combined analytic/numerical method was used by Richardson et al. [5, 18] to reveal how iodide ion vacancies accumulate/deplete in very narrow layers (called Debye layers) adjacent to the perovskite boundaries. The associated rapid change in solution across these Debye layers is a significant source of spatial stiffness, while the disparity in timescales between the motion of electronic and ionic charges is a source of temporal stiffness [6], rendering the task of finding solutions to realistic models of PSCs very challenging.

Courtier et al. [6] developed a finite element scheme, implemented in MATLAB [14], that is able to cope with the spatial and temporal stiffness of the problem and obtain

Electronic supplementary material The online version of this article (<https://doi.org/10.1007/s10825-019-01396-2>) contains supplementary material, which is available to authorised users.

✉ N. E. Courtier
N.E.Courtier@soton.ac.uk

¹ Mathematical Sciences, University of Southampton, Southampton, UK

² Department of Physics, University of Bath, Bath, UK

³ School of Mathematics and Physics, University of Portsmouth, Portsmouth, UK

accurate solutions to a coupled model for ion migration and charge carrier transport within the perovskite layer of a PSC. Since then, alternative numerical methods have also appeared in [13, 27]. Walter et al. [27] have developed a solver that includes the motion of both cations and anions. Their scheme is based on the freely available software Quokka3 [10], originally designed for the solution of models of silicon-based photovoltaic devices. While this provides a thoroughly tested and validated framework for their results, the model being solved only explicitly models the perovskite absorber and there is strong evidence suggesting that the adjacent transport layers play a key role in determining device behaviour [4]. Meanwhile, Jacobs et al. [13] use the proprietary COMSOL package via a MATLAB interface to simulate three layers of a PSC over a range of timescales. However, the details of the modelling and solutions techniques are not given in full, making their results difficult to reproduce and compare with alternatives.

In this work, we present the extension of the finite element code presented in [6] for a single-layer model of a PSC to a model that explicitly describes the three core layers of a PSC: the electron transport layer (ETL), perovskite absorber layer and hole transport layer (HTL). Full details of the charge transport model equations and the implementation of the code are given. The code is freely available on GitHub at <https://github.com/PerovskiteSCModelling/IonMonger> (under an AGPL-3.0 copyleft license) and can be used to simulate a variety of different experimental protocols. Current density–voltage (J – V) curves are the typical measurement that is performed to assess solar cell performance including, for perovskite solar cells, the extent of J – V hysteresis displayed at a particular scan rate. In addition to J – V curves, the code can be used to simulate photocurrent transients (during which the applied voltage and/or the illumination intensity is varied over time) and photovoltage transients (during which the cell is held at open circuit and the illumination intensity is varied) that occur on timescales of microseconds to minutes. Uncovering the links between model parameters and the results of such simulations will help to improve understanding of the underlying physics of PSCs and hence guide further improvements in their design. One such investigation, into how the extent of observable hysteresis depends on material properties of the two transport layers, has been conducted by Courtier et al. [4]. The investigation is based upon simulations of the same¹ three-layer model as that considered in this work.

¹ The simulations in [4] are based on the same mathematical model as described in this work; however, the definitions of the constants of proportionality k_E and k_H and the built-in voltage V_{bi} vary between the two works; the definitions in this work are compatible with Boltzmann statistics for non-degenerate semiconductors, see (13)–(17).

In Sect. 2, we show an example set of simulation results obtained using the code. In Sect. 3, we present and discuss the governing equations in each of the three layers as well as the boundary and interface conditions through which the equations couple together. A non-dimensionalisation is also presented which is geared to study the behaviour of the cell on the timescales associated with anion vacancy motion and aids in obtaining uniformly well-resolved solutions by ensuring that the numerical tolerances are applied equally to each of the model variables. In Sect. 4, we detail the finite element discretisation of the system and highlight the differences between how open-circuit and applied voltage protocols are imposed at the discrete level. The focus of Sect. 5 is a discussion of the how the time-stepping is carried out and how the output current density is calculated from the numerical solution. In Sect. 6, we validate the results of the numerical scheme upon which `IonMonger` is based. Finally, in Sect. 7, we draw our conclusions.

2 Application example

The main purpose of this paper is to provide the perovskite solar cell community with a high-quality, free and useful tool with which to better understand PSC behaviour, and not to provide a detailed analysis or interpretation of the device physics. The power of the solver in advancing our physical understanding of PSCs is demonstrated in Courtier et al. [4], in which an investigation of the effects of material properties of the transport layers on cell performance is detailed. There it is found that two material properties in particular, namely the permittivity and the effective doping density of the transport layers, have a significant role to play in determining the extent of J – V hysteresis exhibited by a PSC. In addition, characteristics of simulations that can be used to identify the dominant recombination mechanism in a PSC are discussed. Results from two other simulations, also computed using the capabilities offered by `IonMonger`, have been presented by Idígoras et al. [12] in a study of the role of surface (also often termed interfacial or interface) recombination, which occurs on the interfaces between the perovskite layer and the adjacent transport layers, on PSC performance. However, further work in this area is vital for the future development and optimisation of PSCs. Here, we show how `IonMonger` can be used to both simulate the most common measurement protocol for assessing the performance of a solar cell, namely a J – V curve, and reveal how each type of recombination included in the model contributes to the observed behaviour.

Figure 1a shows the J – V output for an example simulation of a typical J – V measurement protocol performed on a planar, standard-architecture PSC (see Fig. 2 for the cell geometry). The cell is initially preconditioned for 5 s

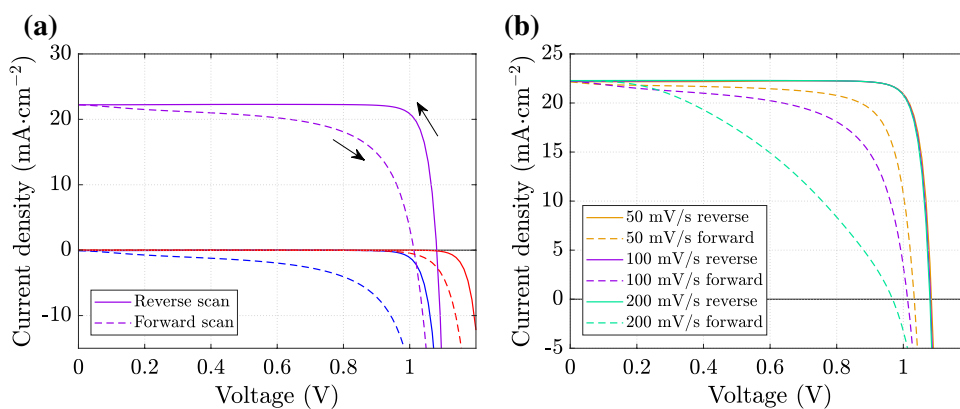


Fig. 1 Example simulation results: **a** A J - V curve measured at 100 mV/s from 1.2 V to short-circuit and back, after a 5-s preconditioning step. The simulation uses the parameters given in Tables 1 and 2(b) of [4] except that here the effective doping densities $d_E = d_H = 10^{24} \text{ m}^{-3}$ and the effective densities of state $g_c^E = g_v^H = 5 \times 10^{24} \text{ m}^{-3}$. The purple lines show the current-density output, while the blue and red lines show the current losses due to interface recombination at the ETL/perovskite and perovskite/HTL interfaces, respectively. Losses due to bulk recombination are not

shown. The direction of scan is indicated by both the arrows and the style of each line: solid for the reverse scan and dashed for the subsequent forward scan. **b** A set of three J - V curves. Here, the example in panel (a) is plotted alongside two other J - V curves measured at scan rates of 50 mV/s and 200 mV/s, but for otherwise identical input parameters. These results demonstrate the ability of the model to reproduce the scan rate-dependent J - V hysteresis commonly exhibited by PSCs due to the migration of ion vacancies within the perovskite layer (Color figure online)

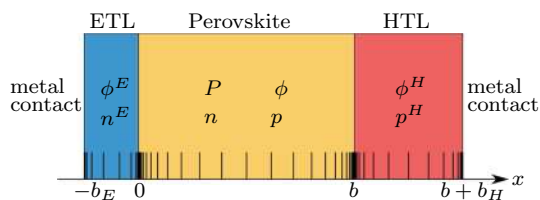


Fig. 2 A schematic of the three-layer geometry of the PSC model detailed in Sect. 3. The symbols in each layer indicate which variables are explicitly modelled in that region. The markers along the x -axis represent the non-uniform grid spacing used by IonMonger

before the applied voltage is scanned from 1.2 V to short circuit (0 V) and back to 1.2 V at a scan rate of 100 mV/s. In addition, Fig. 1a displays the corresponding current losses due to the two types of interface recombination included in the model. Note that bulk recombination occurring within the perovskite layer is also included in the model but is not shown. By comparing the shape of the current-loss curves to the J - V curve, it is clear that, for this example, the observed behaviour is controlled primarily by the rate of recombination at the ETL/perovskite interface (the blue line), while the rate of recombination at the perovskite/HTL interface (the red line) has little effect on the performance of the cell. The parameter values used in the simulation are equal to those given in Tables 1 and 2(b) of [4] except that here the effective doping densities $d_E = d_H = 10^{24} \text{ m}^{-3}$ and the effective densities of state $g_c^E = g_v^H = 5 \times 10^{24} \text{ m}^{-3}$. The input file for this simulation, along with a GUIDE and documentation to aid in using the code, is provided in the main folder of the

IonMonger GitHub repository so that users can utilise these as a starting point for investigations of their own.

In Fig. 1b, the example J - V curve from panel (a) is shown alongside the corresponding results for two other simulations. The only difference in the input parameters between the three simulations is the rate at which the applied voltage is scanned back and forth to measure the J - V curve. The three scan rates are 50, 100 and 200 mV/s. Harvesting the full set of results in Fig. 1b required a total of 34 s of computation time on a standard desktop machine, i.e. approximately 11 s per simulation including the calculation of appropriate initial conditions and the preconditioning step. See Sect. 6 for figures showing how the accuracy of the solution and the computation time vary with respect to the number of grid points on which the solution is computed. Next, the equations that underlie these simulations are detailed.

3 The charge transport model

In this section, the charge transport model for a planar lead halide perovskite solar cell consisting of a perovskite absorber layer sandwiched between an electron transport layer (ETL) and a hole transport layer (HTL) is presented. Tables of the model variables and parameters along with their definitions are given in the SI. The structure of the cell is shown in Fig. 2. The non-dimensionalisation used by the code is also given.

3.1 Model equations

Perovskite absorber layer ($0 < x < b$) A model for the perovskite layer consists of equations for the conservation of conduction (free) electrons, holes and halide ion vacancies coupled with the Poisson equation for the electric potential, $\phi(x, t)$. We denote the halide ion vacancy density by $P(x, t)$, its flux by $F^P(x, t)$ and define \hat{N}_0 as the mean ion vacancy density. It is assumed that an equal, uniform density \hat{N}_0 of immobile cation vacancies also exists within the perovskite layer. The electron and hole concentrations are denoted by n and p with current densities j^n and j^p , respectively. The functions $G(x, t)$ and $R(n, p)$ denote the charge carrier photogeneration and bulk recombination rates, respectively. In the perovskite layer, we thus have

$$\begin{aligned} \frac{\partial p}{\partial t} + \frac{1}{q} \frac{\partial j^p}{\partial x} &= G(x, t) - R(n, p), \\ j^p &= -qD_p \left(\frac{\partial p}{\partial x} + \frac{p}{V_T} \frac{\partial \phi}{\partial x} \right), \end{aligned} \quad (1)$$

$$\begin{aligned} \frac{\partial n}{\partial t} - \frac{1}{q} \frac{\partial j^n}{\partial x} &= G(x, t) - R(n, p), \\ j^n &= qD_n \left(\frac{\partial n}{\partial x} - \frac{n}{V_T} \frac{\partial \phi}{\partial x} \right), \end{aligned} \quad (2)$$

$$\begin{aligned} \frac{\partial P}{\partial t} + \frac{\partial F^P}{\partial x} &= 0, \\ F^P &= -D_1 \left(\frac{\partial P}{\partial x} + \frac{P}{V_T} \frac{\partial \phi}{\partial x} \right), \end{aligned} \quad (3)$$

with Poisson's equation,

$$\frac{\partial^2 \phi}{\partial x^2} = \frac{q}{\epsilon_A} (\hat{N}_0 - P + n - p). \quad (4)$$

Here, D_1 denotes the diffusion coefficient of the iodide ion vacancies and ϵ_A is the permittivity of the perovskite absorber layer. These differential equations are supplemented by continuity conditions at the interfaces with the transport layers (given at the end of this section).

Electron transport layer ($-b_E < x < 0$) The majority carriers through the ETL are free electrons. The model for the electrical behaviour of this layer thus consists only of a conservation equation for the free electrons which couples to Poisson's equation. Here, D_E denotes the electron diffusion coefficient, ϵ_E the permittivity and d_E the intrinsic free electron density (due to the doping) in the ETL.

$$\frac{\partial n}{\partial t} - \frac{1}{q} \frac{\partial j^n}{\partial x} = 0, \quad j^n = qD_E \left(\frac{\partial n}{\partial x} - \frac{n}{V_T} \frac{\partial \phi}{\partial x} \right), \quad (5)$$

$$\frac{\partial^2 \phi}{\partial x^2} = \frac{q}{\epsilon_E} (n - d_E). \quad (6)$$

These equations couple to the perovskite equations via four continuity conditions at the interface (given at the end of this section). On the external boundary with the metal contact, we impose Ohmic boundary conditions. These read

$$n = d_E, \quad \phi = \frac{V_{bi} - V(t)}{2} \quad \text{on } x = -b_E, \quad (7)$$

where $V(t)$ is the applied voltage and V_{bi} denotes the cell's built-in voltage, which is defined in (17).

Hole transport layer ($b < x < b + b_H$) The majority carriers in the HTL are holes, and analogously to the case in the ETL, we need specify only two equations, specifically

$$\frac{\partial p}{\partial t} + \frac{1}{q} \frac{\partial j^p}{\partial x} = 0, \quad j^p = -qD_H \left(\frac{\partial p}{\partial x} + \frac{p}{V_T} \frac{\partial \phi}{\partial x} \right), \quad (8)$$

$$\frac{\partial^2 \phi}{\partial x^2} = \frac{q}{\epsilon_H} (d_H - p). \quad (9)$$

Here, D_H is the hole diffusion coefficient, ϵ_H is the permittivity and d_H is the intrinsic hole density (due to the doping) in the HTL. These equations couple to the equations in the perovskite via four continuity conditions at the interface (given at the end of this section) and satisfy the following Ohmic contact conditions on the metal contact.

$$p = d_H, \quad \phi = -\frac{V_{bi} - V(t)}{2} \quad \text{on } x = b + b_H. \quad (10)$$

Continuity conditions on the interfaces ($x = 0$ and $x = b$) At the interface between the perovskite and the ETL, (1) the electron flux (and its associated current density) is conserved, (2) the hole flux (and its associated current density) is conserved, (3) there is no flux of halide ion vacancies, (4) and (5) both the electrostatic potential and electric displacement field are continuous, and (6) the majority carrier density (in this case the electrons) at the edge of the ETL is related to the neighbouring carrier density in the perovskite by a factor, k_E , which depends upon the relevant band offset and change in effective density of states [11]. Therefore, at the interface between the ETL and the perovskite, the following conditions are applied

$$\left. \begin{aligned} j^n|_{x=0^-} &= j^n|_{x=0^+} - \bar{R}_1 \\ j^p &= -q\bar{R}_1 \\ F^P &= 0 \\ \phi|_{x=0^-} &= \phi|_{x=0^+} \\ \epsilon_E \frac{\partial \phi}{\partial x} \Big|_{x=0^-} &= \epsilon_A \frac{\partial \phi}{\partial x} \Big|_{x=0^+} \\ k_E n|_{x=0^-} &= n|_{x=0^+} \end{aligned} \right\} \text{on } x = 0. \quad (11)$$

Analogous conditions are applied at the interface between the perovskite and the HTL (where the holes are the majority carrier). These read

$$\left. \begin{aligned} j^p|_{x=b^-} - \bar{R}_r &= j^p|_{x=b^+} \\ j^n &= -q\bar{R}_r \\ F^P &= 0 \\ \phi|_{x=b^-} &= \phi|_{x=b^+} \\ \epsilon_A \frac{\partial \phi}{\partial x} \Big|_{x=b^-} &= \epsilon_H \frac{\partial \phi}{\partial x} \Big|_{x=b^+} \\ p|_{x=b^-} &= k_H p|_{x=b^+} \end{aligned} \right\} \text{ on } x = b. \tag{12}$$

Here, the superscripts of \pm denote quantities evaluated at either the left- or right-hand side of the perovskite/transport layer interfaces, respectively; \bar{R}_l and \bar{R}_r are the recombination fluxes at the left (ETL/perovskite) and right (perovskite/HTL) interface, respectively; and k_E and k_H are constants of proportionality between the charge carrier concentrations on each side of the interfaces according to Boltzmann statistics, given by

$$k_E = \frac{g_c}{g_c^E} \exp\left(-\frac{E_c - E_c^E}{k_B T}\right), \tag{13}$$

$$k_H = \frac{g_v}{g_v^H} \exp\left(\frac{E_v - E_v^H}{k_B T}\right). \tag{14}$$

In these expressions, $g_{c,v}$ denote the effective conduction/valence band density of states; $E_{c,v}$ are the energies of the conduction/valence band edges; and the superscripts E or H indicate to which transport layer a quantity relates. The validity of each of these expressions relies on the validity of using the Boltzmann approximation to describe the distribution of electrons in a non-degenerate semiconductor. Consequently, users should choose an effective doping density which is less than 20 times smaller than the effective density of states in each transport layer in order to ensure that the equilibrium Fermi level is more than a few thermal voltages away from the band edges, see (15)–(16).

Built-in voltage The cell’s built-in voltage is equal to the difference between the workfunctions of the two metal contacts. Assuming that the contacts form ideal Ohmic contacts with the adjacent transport layer, this difference is equal to the difference between the equilibrium Fermi levels of the two transport layers which are approximated, using Boltzmann statistics, by

$$E_f^E = E_c^E - k_B T \log\left(\frac{g_c^E}{d_E}\right), \tag{15}$$

$$E_f^H = E_v^H + k_B T \log\left(\frac{g_v^H}{d_H}\right). \tag{16}$$

Hence, the built-in voltage is given by

$$V_{bi} = E_c^E - E_v^H - k_B T \log\left(\frac{g_c^E g_v^H}{d_E d_H}\right). \tag{17}$$

3.2 Carrier generation and recombination rates

For the rate of charge carrier generation within the perovskite layer, we use a simplified Beer–Lambert model of light absorption [16] in which it is assumed that absorption can be characterised by a single absorption coefficient (α) and photon flux that are independent of the wavelength of light. Taking into account the possibility of choosing either a standard or inverted architecture (i.e. applying the light to either the ETL- or HTL-side of the cell, respectively), this rate can be written as

$$G(x, t) = I_s(t) F_{ph} \alpha \exp\left(-\alpha \left[\frac{b}{2} + l\left(x - \frac{b}{2}\right)\right]\right), \tag{18}$$

in which F_{ph} denotes the flux of photons incident on the light-facing perovskite surface (after accounting for reflection) under the equivalent of 1 Sun illumination; the function $I_s(t)$ is the intensity of the illumination in Sun equivalents; and the parameter l can be set equal to either +1 for light from the left (through the ETL) or -1 for light from the right (through the HTL) by making use of the `Inverted` option in the parameter input file.

We allow bulk recombination to be described by a combination of bimolecular and trap-assisted Shockley–Read–Hall (SRH) recombination mechanisms from, for example, Nelson [16] §4.5.5. Hence, the volumetric bulk recombination rate is

$$R(n, p) = \beta(np - n_i^2) + \frac{np - n_i^2}{\tau_n(p + p_t) + \tau_p(n + n_t)}, \tag{19}$$

in which β is the bimolecular rate constant, n_i is the intrinsic carrier concentration, τ_n and τ_p are the charge carrier lifetimes and we assume that the trap state energy level lies close to the intrinsic potential energy of the perovskite such that we can apply the approximation that $p_t = n_t = n_i$.

Similarly, the interfacial recombination fluxes (\bar{R}_l and \bar{R}_r) can be chosen as a combination of bimolecular and SRH recombination as follows, noting the use of recombination velocities rather than carrier lifetimes in the SRH recombination rates (see Nelson [16, Sect. 4.5.6]),

$$\bar{R}_{1,r}(n^-, p^+) = \beta^{E,H}(n^- p^+ - n_i^- n_i^+) + \frac{n^- p^+ - n_i^- n_i^+}{\frac{1}{v_n^{E,H}}(p^+ + p_i^+) + \frac{1}{v_p^{E,H}}(n^- + n_i^-)}, \tag{20}$$

in which the superscripts of \pm denote quantities evaluated at either the left- or right-hand side of the perovskite/transport layer interfaces, respectively; $\beta^{E,H}$ are the bimolecular rate constants; and $v_n^{E,H}$ and $v_p^{E,H}$ are the electron and hole recombination velocities, respectively. In order to satisfy the principle of detailed balance, we assume that the intrinsic carrier density in the ETL $n_i^E = \frac{n_i}{k_E}$ and the intrinsic carrier density in the HTL $n_i^H = \frac{n_i}{k_H}$ for consistency with the continuity conditions in (11f) and (12f). Then, in order to keep the number of input parameters to a minimum, we approximate $p_i = n_i = n_i$ in analogy with the approximation made to the bulk recombination rate above. Therefore, the interface recombination rates used by the code are equivalent to

$$\bar{R}_1\left(\frac{n}{k_E}, p\right) = \frac{\beta^E}{k_E}(np - n_i^2) + \frac{np - n_i^2}{\frac{k_E}{v_n^E}(p + n_i) + \frac{1}{v_p^E}(n + n_i)} \quad \text{on } x = 0^+, \tag{21}$$

$$\bar{R}_r\left(n, \frac{p}{k_H}\right) = \frac{\beta^H}{k_H}(np - n_i^2) + \frac{np - n_i^2}{\frac{1}{v_n^H}(p + n_i) + \frac{k_H}{v_p^H}(n + n_i)} \quad \text{on } x = b^-. \tag{22}$$

3.3 Calculation of the total current density

In order to calculate the total current density from a numerical solution of the drift–diffusion model, we derive an expression that can be evaluated at any point in the domain. The code automatically calculates the current density at the midpoint of the perovskite layer, where the grid spacing is larger and the solution varies more smoothly than in the Debye layers, to minimise numerical error. By subtracting Eq. (2a) from Eq. (1a), we get

$$\frac{\partial}{\partial t}(p - n) + \frac{1}{q} \frac{\partial}{\partial x}(j^p + j^n) = 0. \tag{23}$$

Then, by substituting the difference in the carrier concentrations ($p - n$) using Poisson’s equation for the perovskite layer, given in (4), and multiplying by q , we get

$$\frac{\partial}{\partial x}(j^p + j^n) - \frac{\partial}{\partial t}\left(\epsilon_A \frac{\partial^2 \phi}{\partial x^2} - q(\hat{N}_0 - P)\right) = 0. \tag{24}$$

Applying the time derivative to the last term in the brackets allows us to eliminate \hat{N}_0 (the constant and uniform cation vacancy density) and to use the ion vacancy conservation equation in (3) to make a substitution for the time derivative of P , which gives

$$\frac{\partial}{\partial x}(j^p + j^n) - \frac{\partial}{\partial t}\left(\epsilon_A \frac{\partial^2 \phi}{\partial x^2}\right) + q \frac{\partial F^P}{\partial x} = 0. \tag{25}$$

Similarly, for the transport layers, we have

$$\frac{\partial j^n}{\partial x} - \frac{\partial}{\partial t}\left(\epsilon_E \frac{\partial^2 \phi}{\partial x^2}\right) = 0, \tag{26}$$

$$\frac{\partial j^p}{\partial x} - \frac{\partial}{\partial t}\left(\epsilon_H \frac{\partial^2 \phi}{\partial x^2}\right) = 0. \tag{27}$$

After swapping the order of spatial and temporal differentiation, it is possible to integrate these three equations with respect to the spatial variable x . By integrating and applying the continuity conditions and the Ohmic boundary conditions at either metal contact, we get an expression for the total current density which is independent of x and given by

$$J(t) = \begin{cases} j^n + j^p - \frac{\partial}{\partial t}\left(\epsilon_A \frac{\partial \phi}{\partial x}\right) + qF^P, & \text{for } 0 < x < b, \\ j^n - \frac{\partial}{\partial t}\left(\epsilon_E \frac{\partial \phi}{\partial x}\right), & \text{for } -b_E < x < 0, \\ j^p - \frac{\partial}{\partial t}\left(\epsilon_H \frac{\partial \phi}{\partial x}\right), & \text{for } b < x < b + b_H. \end{cases} \tag{28}$$

The term involving a time derivative in each equation is the displacement current density. It should be noted that the magnitude of the displacement current density is usually even smaller than the magnitude of the expected numerical error in the solution, see Sect. 6.

3.4 Boundary conditions for open-circuit conditions

It is possible to simulate open-circuit conditions, rather than a fixed voltage protocol, by changing just two of the model equations. The two conditions that describe the application of a fixed potential difference are (7b) and (10b). The model for simulation of a device at open circuit instead includes boundary conditions to ensure that there is zero flux of electrons across the metal/ETL boundary (and hence no photocurrent) and that the values of the electric potential at each contact are equal and opposite. This amounts to imposing, in place of (7b) and (10b),

$$j^n|_{x=-b_E} = 0, \quad \phi|_{x=-b_E} = -\phi|_{x=b+b_H}. \tag{29}$$

3.5 Non-dimensionalisation

The code is programmed to solve a non-dimensional form of the model equations. Note, however, that all input parameter values are non-dimensionalised automatically and the output is re-dimensionalised prior to output. Details of the non-dimensionalisation are given here to allow the possibility that users can adapt the equations that underlie the model described here. The non-dimensionalisation is given by:

$$\begin{aligned} x &= bx^*, & t &= \tau_{\text{ion}}t^*, & \phi &= V_T\phi^*, \\ p &= d_H p^*, & n &= d_E n^*, & P &= \hat{N}_0 P^*, \\ j^p &= qG_0 b j^{p*}, & j^n &= qG_0 b j^{n*}, & F^P &= \frac{D_1 \hat{N}_0}{b} F^{P*}, \end{aligned} \tag{30}$$

where G_0 is a typical value of G (the rate of photogeneration of charge pairs per unit volume) and τ_{ion} is the characteristic timescale for ion motion into the Debye layers, given by

$$G_0 = \frac{F^{\text{ph}}}{b} (1 - e^{-ab}), \quad \tau_{\text{ion}} = \frac{b}{D_1} \sqrt{\frac{V_T \epsilon_A}{q \hat{N}_0}}. \tag{31}$$

The perovskite (ionic) Debye length is defined as

$$L_D = \sqrt{\frac{V_T \epsilon_A}{q \hat{N}_0}}. \tag{32}$$

The other input functions and constants are rescaled as follows:

$$\begin{aligned} V &= V_T \Phi, & V_{\text{bi}} &= V_T \Phi_{\text{bi}}, & G &= G_0 G^*, \\ R &= G_0 R^*, & \bar{R}_1 &= b G_0 \bar{R}_1^*, & \bar{R}_r &= b G_0 \bar{R}_r^*, \\ \sigma &= \frac{d_E}{G_0 \tau_{\text{ion}}}, & \chi &= \frac{d_H}{d_E}, & \kappa_p &= \frac{D_p d_H}{G_0 b^2}, \\ \kappa_n &= \frac{D_n d_E}{G_0 b^2}, & \kappa_E &= \frac{D_E \kappa_n}{D_n}, & \kappa_H &= \frac{D_H \kappa_p}{D_p}, \\ w_E &= \frac{b_E}{b}, & w_H &= \frac{b_H}{b}, & r_E &= \frac{\epsilon_E}{\epsilon_A}, \\ r_H &= \frac{\epsilon_H}{\epsilon_A}, & \delta &= \frac{d_E}{\hat{N}_0}, & \lambda &= \frac{L_D}{b}, \\ \lambda_E &= \sqrt{\frac{\epsilon_E \hat{N}_0}{\epsilon_A d_E}} \lambda, & \lambda_H &= \sqrt{\frac{\epsilon_H \hat{N}_0}{\epsilon_A d_H}} \lambda. \end{aligned} \tag{33}$$

The star notation is dropped in the following sections.

4 Discretisation

The numerical method upon which our code is based was developed by Courtier et al. [6] to solve a simplified model description of a perovskite solar cell in which it was assumed that the transport layers were so highly doped that the potential within them was uniform, thereby reducing the

model to equations in the perovskite absorber layer only. In that work, the speed and accuracy of the method are compared against those of two previously used alternatives. It is shown that the method we adopt here is superior to both of these methods for both metrics of performance. Here, we adapt the finite element-based scheme to solve the dimensionless three-layer model set out in the previous section. We do not use the Scharfetter–Gummel scheme [19], commonly used for the solution of drift–diffusion equations, because it is tailored to deal with issues related to charge carrier transport rather than accurately resolving solutions in narrow Debye layers, which is the main difficulty in the present work.

4.1 Spatial grid

The discretisation is formulated on a computational grid comprised of $N + N_E + N_H + 1$ non-uniformly positioned grid points which partition the (non-dimensional) domain $x \in [-w_E, 1 + w_H]$ into $N + N_E + N_H$ subintervals. The perovskite layer (including interfaces) contains $N + 1$ grid points denoted by $x = x_i$ for $i = 0, \dots, N$ with subinterval widths denoted by $\Delta_{i+1/2} = x_{i+1} - x_i$. The transport layer domains (excluding interfaces) contain N_E and N_H grid points, respectively, with grid points denoted by $x = x_i^E$ for $i = 0, \dots, N_E - 1$ and $x = x_i^H$ for $i = 1, \dots, N_H$, respectively, with corresponding subinterval notation.

It is known that the largest gradients in the solution appear in narrow Debye layers adjacent to the material interfaces [5]. This motivates the use of a grid in which points are concentrated near the domain boundaries and at internal interfaces so that computational resolution is focused there and not wasted where it is not required. One such grid can be created by extending the *tanh* grid used in [6] to the three-layer cell geometry displayed in Fig. 2. Specifically, we set

$$x_i = \frac{1}{2} \left(\frac{\tanh \left[s \left(\frac{2i}{N} - 1 \right) \right]}{\tanh(s)} + 1 \right), \tag{34}$$

for $i = 0, \dots, N$,

$$x_i^E = \frac{w_E}{2} \left(\frac{\tanh \left[s \left(\frac{2i}{N_E} - 1 \right) \right]}{\tanh(s)} - 1 \right), \tag{35}$$

for $i = 0, \dots, N_E - 1$,

$$x_i^H = 1 + \frac{w_H}{2} \left(\frac{\tanh \left[s \left(\frac{2i}{N_H} - 1 \right) \right]}{\tanh(s)} + 1 \right), \tag{36}$$

for $i = 1, \dots, N_H$.

Our numerical experiments indicate that a good rule of thumb for deciding on a judicious choice for the value for s can be to calculate a value which leads to 20% of the grid points falling within one Debye length of each interface within the perovskite layer (i.e. the intervals $x \in [0, \lambda]$ and $x \in [1 - \lambda, 1]$), via numerical solution of

$$\lambda = \frac{1}{2} \left(\frac{\tanh(0.6)}{\tanh(s)} + 1 \right). \tag{37}$$

The code is set up so that the numbers of grid points N_E and N_H are chosen based on the input parameter N to give approximately equal spacing immediately either side of the interface.

4.2 Finite element scheme

As in [6], we employ a common finite element approach, in which each of the dependent variables is approximated as a linear combination of piecewise linear basis functions with compact support. For a generic dependent variable, u , defined within the perovskite, i.e. for $x \in (0, 1)$, we write

$$u(x, t) = \sum_{i=0}^{i=N} u_i(t) \varphi_i(x) \tag{38}$$

where

$$\varphi_i(x) = \begin{cases} \frac{x-x_{i-1}}{x_i-x_{i-1}} & \text{if } x \in (x_{i-1}, x_i) \\ \frac{x_{i+1}-x}{x_{i+1}-x_i} & \text{if } x \in (x_i, x_{i+1}) \\ 0 & \text{if } x \notin (x_{i-1}, x_{i+1}) \end{cases} \tag{39}$$

in which $\varphi_i(x)$ are referred to as the basis functions. Each of the governing equations of interest can be manipulated into the form

$$A \frac{\partial u}{\partial t} = B \frac{\partial}{\partial x} \left(\frac{\partial u}{\partial x} \pm u \frac{\partial \phi}{\partial x} \right) + S(x, t, u, v_1, v_2, v_3), \tag{40}$$

in which A and B are constants and the function $S(x, t, u, v_1, v_2, v_3)$ is a source term which depends upon the spatial variable x , the temporal variable t , the generic variable u and a series of other generic dependent variables v_i for $i = 1, 2, 3$. The electron, hole and halide ion vacancy conservation equations, (1), (2) and (3), are rewritten in this

form by eliminating the attendant electron or hole current densities, or the halide ion vacancy flux, respectively. Poisson’s equation in the perovskite, (4), is already in this form.

The spatially discretised equations in the perovskite are derived by multiplying (40) through each of the test functions $\varphi_j(x)$ (for $j = 0, \dots, N$), integrating over the domain $x \in (0, 1)$ (using integration by parts where appropriate) and substituting form (38) for each of the dependent variables. On doing so, we arrive at

$$\begin{aligned} & A \sum_{i=0}^{i=N} \frac{du_i}{dt} \int_0^1 \varphi_i \varphi_j dx \\ & = B \left(\frac{\partial u}{\partial x} \pm u \frac{\partial \phi}{\partial x} \right) \varphi_j \Big|_{x=0}^{x=1} \\ & - B \left\{ \sum_{i=0}^{i=N} u_i \int_0^1 \varphi_i' \varphi_j' dx \right. \\ & \left. \pm \sum_{i=0}^{i=N} \sum_{k=0}^{k=N} u_i \varphi_k \int_0^1 \varphi_i \varphi_j' \varphi_k' dx \right\} \\ & + \int_0^1 S(x, t, u, v_1, v_2, v_3) \varphi_j dx. \end{aligned} \tag{41}$$

Each of the integrals containing expressions that depend solely on the basis functions and/or their derivatives can be computed exactly. Likewise, terms containing quantities evaluated on the boundaries $x = 0, 1$ can be computed exactly using the continuity conditions (11)–(12), else the relevant equation is replaced by the corresponding Dirichlet condition. The one remaining term that is not so readily computed is the final integral in (41) that depends on the source terms S . For the anion vacancy conservation, $S \equiv 0$ and so this term is zero. For Poisson’s equation, this term is a linear combination of dependent variables and so can be computed exactly. However, for the electron and hole conservation equations, (2) and (1), the source term comprises both the generation and bulk recombination rates, $G(x)$ and $R(n, p)$, which are highly nonlinear, see Sect. 3.2. In order that the integral contained in the final term of (41) can be integrated (at least approximately) regardless of the functional form of the source term, we make a further approximation, that is, to replace the dependent variables in the integrand by functions that are piecewise constant over each subinterval, $x \in (x_i, x_{i+1})$, and have a value equal to that of the full series (38) at the midpoint of that interval. In short, we make the additional approximation

$$\int_0^1 (G - R(n, p))\varphi_j \, dx \approx \begin{cases} \frac{\Delta_{j-\frac{1}{2}}}{2} \left(G|_{x=x_{j-\frac{1}{2}}} - R\left(n|_{x=x_{j-\frac{1}{2}}}, p|_{x=x_{j-\frac{1}{2}}}\right) \right) \\ + \frac{\Delta_{j+\frac{1}{2}}}{2} \left(G|_{x=x_{j+\frac{1}{2}}} - R\left(n|_{x=x_{j+\frac{1}{2}}}, p|_{x=x_{j+\frac{1}{2}}}\right) \right), & \text{if } j = 1, \dots, N-1, \\ \frac{\Delta_{\frac{1}{2}}}{2} \left(G|_{x=x_{\frac{1}{2}}} - R\left(n|_{x=x_{\frac{1}{2}}}, p|_{x=x_{\frac{1}{2}}}\right) \right), & \text{if } j = 0, \\ \frac{\Delta_{N-\frac{1}{2}}}{2} \left(G|_{x=x_{N-\frac{1}{2}}} - R\left(n|_{x=x_{N-\frac{1}{2}}}, p|_{x=x_{N-\frac{1}{2}}}\right) \right), & \text{if } j = N. \end{cases} \quad (42)$$

The additional error incurred as a result of this approximation is comparable to the error associated with the original piecewise linear approximation for the dependent variables. Thus, even though some additional error is introduced, the scheme retains its second-order local accuracy, as demonstrated in Sect. 6. We note that this approach to deal with the nonlinear source terms is a special case of the method used in the work of Skeel and Berzins [22], but we emphasise that in contrast to their method, we only use this additional approximation for treatment of the source terms while the rest of the terms are integrated exactly.

An analogous methodology is used to derive the discrete equations in the transport layers. The only difference is that the basis and test functions are piecewise linear functions with compact support defined within the ETL and HTL, respectively.

For notational convenience, we introduce three discrete operators: a difference operator, \mathfrak{D}_i ; an interpolation operator, \mathfrak{F}_i ; and, a linear operator \mathfrak{L}_i . For a generic dependent variable u , these three operators are defined as follows, in which the midpoint $x = x_{i+1/2} = x_i/2 + x_{i+1}/2$ is halfway between $x = x_i$ and $x = x_{i+1}$.

$$\left. \frac{\partial u}{\partial x} \right|_{x=x_{i+1/2}} \approx \mathfrak{D}_{i+1/2}(u) = \frac{u_{i+1} - u_i}{\Delta_{i+1/2}}, \quad (43)$$

$$u|_{x=x_{i+1/2}} \approx \mathfrak{F}_{i+1/2}(u) = \frac{1}{2}(u_{i+1} + u_i), \quad (44)$$

$$\mathfrak{L}_i(u) = \frac{1}{6}\Delta_{i+1/2}u_{i+1} + \frac{1}{3}(\Delta_{i+1/2} + \Delta_{i-1/2})u_i + \frac{1}{6}\Delta_{i-1/2}u_{i-1}. \quad (45)$$

These discrete operators can be used to obtain discretised versions of the ion flux and carrier current densities in the

perovskite layer, given in dimensional form in (3b), (2b) and (1b), as follows:

$$F^P_{i+1/2} = -[\mathfrak{D}_{i+1/2}(P) + \mathfrak{F}_{i+1/2}(P)\mathfrak{D}_{i+1/2}(\phi)], \quad (46)$$

$$j^n_{i+1/2} = \kappa_n[\mathfrak{D}_{i+1/2}(n) - \mathfrak{F}_{i+1/2}(n)\mathfrak{D}_{i+1/2}(\phi)], \quad (47)$$

$$j^p_{i+1/2} = -\kappa_p[\mathfrak{D}_{i+1/2}(p) + \mathfrak{F}_{i+1/2}(p)\mathfrak{D}_{i+1/2}(\phi)]. \quad (48)$$

for $i = 0, \dots, N-1$. The electron current density in the ETL ($j^{n,E}$) and hole current density in the HTL ($j^{p,H}$) can be expressed in an equivalent way.

The carrier generation and bulk recombination terms are approximated to be linear on each interval, and to take the value at the midpoint, hence we define

$$G_{i+1/2} = G(\mathfrak{F}_{i+1/2}(x), t), \quad (49)$$

$$R_{i+1/2} = R(\mathfrak{F}_{i+1/2}(n), \mathfrak{F}_{i+1/2}(p)). \quad (50)$$

Perovskite absorber layer The discretised equations governing the evolution of the halide ion vacancy density subject to no-flux boundary conditions, corresponding to (3), (11c) and (12c), are

$$\Delta_{1/2} \left[\frac{1}{3} \frac{dP_0}{dt} + \frac{1}{6} \frac{dP_1}{dt} \right] = -\lambda F^P_{1/2}, \quad (51)$$

$$\mathfrak{L}_i \left(\frac{dP}{dt} \right) = -\lambda [F^P_{i+1/2} - F^P_{i-1/2}], \quad \text{for } i = 1, \dots, N-1, \quad (52)$$

$$\Delta_{N-1/2} \left[\frac{1}{6} \frac{dP_{N-1}}{dt} + \frac{1}{3} \frac{dP_N}{dt} \right] = \lambda F^P_{N-1/2}. \quad (53)$$

The discretisation of Poisson’s equation, from (4), becomes

$$0 = \lambda^2 [\mathfrak{D}_{i+1/2}(\phi) - \mathfrak{D}_{i-1/2}(\phi)] + \mathfrak{L}_i(P) - \frac{1}{2} [\Delta_{i+1/2} + \Delta_{i-1/2}] + \delta [\chi \mathfrak{L}_i(p) - \mathfrak{L}_i(n)] \quad (54)$$

for $i = 1, \dots, N-1$.

The conservation equations for the electrons and holes and the carrier current density boundary conditions, corresponding to (1a), (2a), (11b) and (12b), become

$$\sigma \mathfrak{L}_i \left(\frac{dn}{dt} \right) = j^n_{i+1/2} - j^n_{i-1/2} + \frac{\Delta_{i+1/2}}{2} [G_{i+1/2} - R_{i+1/2}] + \frac{\Delta_{i-1/2}}{2} [G_{i-1/2} - R_{i-1/2}], \quad \text{for } i = 1, \dots, N-1, \quad (55)$$

$$\sigma \Delta_{N-1/2} \left[\frac{1}{6} \frac{dn_{N-1}}{dt} + \frac{1}{3} \frac{dn_N}{dt} \right] = -j^n_{N-1/2} + \frac{\Delta_{N-1/2}}{2} [G_{N-1/2} - R_{N-1/2}] - \bar{R}_r(n_N, p_0^H), \tag{56}$$

$$\sigma \chi \Delta_{1/2} \left[\frac{1}{3} \frac{dp_0}{dt} + \frac{1}{6} \frac{dp_1}{dt} \right] = -j^p_{1/2} + \frac{\Delta_{1/2}}{2} [G_{1/2} - R_{1/2}] - \bar{R}_l(n_{N_E}^E, p_0), \tag{57}$$

$$\sigma \chi \mathfrak{Q}_i \left(\frac{dp}{dt} \right) = -j^p_{i+1/2} + j^p_{i-1/2} + \frac{\Delta_{i+1/2}}{2} [G_{i+1/2} - R_{i+1/2}] + \frac{\Delta_{i-1/2}}{2} [G_{i-1/2} - R_{i-1/2}], \text{ for } i = 1, \dots, N-1. \tag{58}$$

Electron transport layer The equations and Dirichlet boundary conditions for the electric potential and the electron density in the ETL, from (5a), (6) and (7), become

$$0 = \phi_0^E - \frac{\Phi - \Phi_{bi}}{2}, \tag{59}$$

$$0 = \lambda_E^2 \left[\mathfrak{D}_{i+1/2}^E(\phi^E) - \mathfrak{D}_{i-1/2}^E(\phi^E) \right] + \frac{1}{2} \left[\Delta_{i+1/2}^E + \Delta_{i-1/2}^E \right] - \mathfrak{Q}_i^E(n^E), \text{ for } i = 1, \dots, N_E - 1, \tag{60}$$

$$0 = n_0^E - 1, \tag{61}$$

$$\sigma \mathfrak{Q}_i^E \left(\frac{dn^E}{dt} \right) = j^n_{i+1/2, E} - j^n_{i-1/2, E}, \text{ for } i = 1, \dots, N_E - 1. \tag{62}$$

Hole transport layer Similarly, for the electric potential and hole density in the HTL from (8a), (9) and (10),

$$0 = \lambda_H^2 \left[\mathfrak{D}_{i+1/2}^H(\phi^H) - \mathfrak{D}_{i-1/2}^H(\phi^H) \right] + \mathfrak{Q}_i^H(p^H) - \frac{1}{2} [\Delta_{i+1/2} + \Delta_{i-1/2}], \text{ for } i = 1, \dots, N_H - 1, \tag{63}$$

$$0 = \phi_{N_H}^H + \frac{\Phi - \Phi_{bi}}{2}, \tag{64}$$

$$\sigma \chi \mathfrak{Q}_i^H \left(\frac{dp^H}{dt} \right) = - \left[j^p_{i+1/2, H} - j^p_{i-1/2, H} \right], \text{ for } i = 1, \dots, N_H - 1, \tag{65}$$

$$0 = p_{N_H}^H - 1. \tag{66}$$

Continuity conditions The carrier relations and continuity of the potential across the interfaces from (11d, f) and (12d, f) are applied directly as follows:

$$\phi_{N_E}^E = \phi_0, \quad k_E n_{N_E}^E = n_0, \quad \phi_N = \phi_0^H, \quad p_N = k_H p_0^H. \tag{67}$$

The continuity of the displacement field and that of the electric potential across the interfaces, from (11e) and (12e), are ensured via

$$0 = \mathfrak{D}_{1/2}(\phi) - r_E \mathfrak{D}_{N_E-1/2}^E(\phi^E) - \frac{\Delta_{1/2}}{\lambda^2} \left(\frac{1}{2} - \frac{P_1}{3} - \frac{P_2}{6} + \delta \left[\frac{n_1}{3} + \frac{n_2}{6} - \frac{p_1}{3} - \frac{p_2}{6} \right] \right) - r_E \frac{\Delta_{N_E-1/2}^E}{\lambda_E^2} \left(\frac{n_{N_E-1}^E}{6} + \frac{n_0}{3k_E} - \frac{1}{2} \right), \tag{68}$$

$$0 = r_H \mathfrak{D}_{1/2}^H(\phi^H) - \mathfrak{D}_{N-1/2}(\phi) - \frac{\Delta_{N-1/2}}{\lambda^2} \left(\frac{1}{2} - \frac{P_{N-1}}{6} - \frac{P_N}{3} + \delta \left[\frac{n_{N-1}}{6} + \frac{n_N}{3} - \frac{p_{N-1}}{6} - \frac{p_N}{3} \right] \right) - r_H \frac{\Delta_{1/2}^H}{\lambda_H^2} \left(\frac{1}{2} - \frac{p_N}{3k_H} - \frac{p_1^H}{6} \right). \tag{69}$$

Interface recombination is included in the equations for the continuity of carrier current densities, corresponding to (11a) and (12a), as follows:

$$\sigma \left(\frac{1}{6} \Delta_{N_E-1/2}^E \frac{dn_{N_E-1}^E}{dt} + \frac{1}{3} \left(\frac{\Delta_{N_E-1/2}^E}{k_E} + \Delta_{1/2} \right) \frac{dn_0}{dt} + \frac{1}{6} \Delta_{1/2} \frac{dn_1}{dt} \right) = - \left[j^n_{N_E-1/2, E} - j^n_{1/2, E} \right] + \frac{\Delta_{1/2}}{2} [G_{1/2} - R_{1/2}] - \bar{R}_l(n_{N_E}^E, p_0), \tag{70}$$

$$\sigma \chi \left(\frac{1}{6} \Delta_{N-1/2} \frac{dp_{N-1}}{dt} + \frac{1}{3} \left(\Delta_{N-1/2} + \frac{\Delta_{1/2}^H}{k_H} \right) \frac{dp_N}{dt} + \frac{1}{6} \Delta_{1/2}^H \frac{dp_1^H}{dt} \right) = j^p_{N-1/2, H} - j^p_{1/2, H} + \frac{\Delta_{N-1/2}}{2} [G_{N-1/2} - R_{N-1/2}] - \bar{R}_r(n_N, p_0^H). \tag{71}$$

4.3 Boundary conditions for modelling open-circuit conditions

In order to simulate open-circuit conditions using the alternative boundary conditions given in (29), we simply replace (59) and (64) by

$$\sigma \Delta_{1/2}^E \left[\frac{1}{3} \frac{dn_0^E}{dt} + \frac{1}{6} \frac{dn_1^E}{dt} \right] = j_{n,1/2}^E, \tag{72}$$

$$0 = \phi_1^E + \phi_{N_H}^H. \tag{73}$$

5 Implementation

In this section, we outline the steps performed by the code. We begin by describing the procedure that is used to integrate forward in time. Next, we outline how the parameters, operating protocol and initial conditions are set. Finally, we outline how `IonMonger` post-processes the results so that quantities of interest, e.g. the dimensional current output, can be extracted and visualised.

5.1 Integration in time using MATLAB's `ode15s`

The system of differential algebraic equations formulated in Sect. 4.2 is evolved forward in time using MATLAB's `ode15s` [20, 21]. A prerequisite for leveraging this algorithm is assembling the state variables into a column vector, $\mathbf{u}(t)$. A significant decrease in computational cost (proportional to the length of $\mathbf{u}(t)$ squared) is available if the size of $\mathbf{u}(t)$ can be reduced, and so we eliminate superfluous variables, namely F^P , j^n , j^p and E between equations (46)–(73) before assembling the $4N + 2N_E + 2N_H + 4$ remaining unknown functions of time, into the column vector $\mathbf{u}(t)$ as follows:

$$\mathbf{u}(t) = \begin{bmatrix} P_0, \dots, P_N, \phi_0, \dots, \phi_N, \\ n_0, \dots, n_N, p_0, \dots, p_N, \\ \phi_0^E, \dots, \phi_{N_E-1}^E, n_0^E, \dots, n_{N_E-1}^E, \\ \phi_1^H, \dots, \phi_{N_H}^H, p_1^H, \dots, p_{N_H}^H \end{bmatrix}^T \tag{74}$$

$$= \begin{bmatrix} \mathbf{P}(t)^T & \mathbf{\Phi}(t)^T & \mathbf{n}(t)^T & \mathbf{p}(t)^T \\ \mathbf{\Phi}^E(t)^T & \mathbf{n}^E(t)^T & \mathbf{\Phi}^H(t)^T & \mathbf{p}^H(t)^T \end{bmatrix}^T, \tag{75}$$

where a superscript T denotes a transpose. In (75), \mathbf{P} , $\mathbf{\Phi}$, \mathbf{n} and \mathbf{p} are column vectors of length $N + 1$; $\mathbf{\Phi}^E$ and \mathbf{n}^E are column vectors of length N_E ; and $\mathbf{\Phi}^H$ and \mathbf{p}^H are column

vectors of length N_H . The problem to be solved can now be written in the form

$$\mathbf{M} \frac{d\mathbf{u}}{dt} = \mathbf{f}(\mathbf{u}) \quad \text{with} \quad \mathbf{u}|_{t=0} = \mathbf{u}_0, \tag{76}$$

in which $\mathbf{f}(\mathbf{u})$ is a nonlinear vector function of length $4N + 2N_E + 2N_H + 4$ whose entries are the right-hand sides of (51)–(71) and \mathbf{M} is a singular diagonal mass matrix whose entries are the coefficients of the time derivative terms in the same equations.

Another useful strategy for speeding up computations, and one that we make use of in `IonMonger`, is to exploit `ode15s`'s `jpattern` option. This facilitates additional savings in computational cost by specifying entries in the Jacobian of the vector function \mathbf{f} which are known to always equal zero a priori, thereby preventing the algorithm from having to numerically approximate their value as the integration in time proceeds. The function `Jac` creates a sparse matrix that indicates which entries of the Jacobian need to be numerically approximated at each time step and which are always equal to zero.

5.2 Parameter input and initial conditions

The necessary dimensional parameters, the illumination protocol $G(x, t)$, voltage protocol $V(t)$ and solver options are passed between functions in a MATLAB structure called `params`. A `params` structure can most easily be created using the script called `parameters.m`.

Finding initial conditions which satisfy the requisite boundary conditions is non-trivial. In the code, we opt to supply initial conditions that correspond to a device which has been left to reach a quasi-equilibrium with the applied voltage held equal to either a fixed value or the open-circuit voltage such that there is no output current from the cell. In either case, the task of finding initial conditions amounts to finding a valid steady-state solution to the PSC model and we start by finding initial conditions corresponding to when the applied voltage is set equal to the built-in voltage, as defined in (17). This task is tackled by the function `initial_conditions.m`. In order to start the simulation protocol from a different value of the applied voltage or from open-circuit conditions, an additional call is then made to either `precondition.m` or `find_Voc.m`, respectively. All three of these routines find discrete representations of the (dimensionless) initial conditions which can be written as

$$P|_{t=0} = \hat{P}(x), \quad \text{for } 0 \leq x \leq 1, \tag{77}$$

$$\phi|_{t=0} = \hat{\phi}(x), \quad \text{for } -w_E \leq x \leq 1 + w_H, \tag{78}$$

$$n|_{t=0} = \hat{n}(x), \quad \text{for } -w_E \leq x \leq 1, \tag{79}$$

$$p|_{t=0} = \hat{p}(x), \quad \text{for } 0 \leq x \leq 1 + w_H, \tag{80}$$

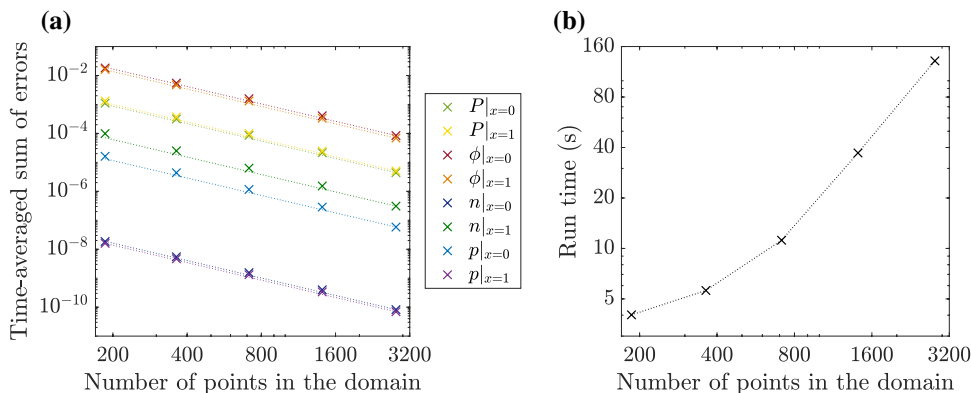


Fig. 3 **a** A measure of the error in each of eight solution variables (listed in the legend) for five simulations versus the number of grid points in the domain on a logarithmic scale. Dotted lines show the expected rate of convergence aligned to the rightmost data point. **b** The computation time for each of the same five simulations versus the number of grid points on a logarithmic scale. Here, a dotted line

is shown as a guide for the eye. For each of the five simulations performed for both panels **a** and **b**, the input parameter N equals 100, 200, 400, 800 and 1600, while the other parameters are identical to those used for Fig. 1a (for which $N = 400$). The number of grid points is equal to $N + N_E + N_H + 1$ where N_E and N_H are automatically computed from N (see Sect. 4.1) (Color figure online)

where the initial profiles $\hat{P}(x)$, $\hat{\phi}(x)$, $\hat{n}(x)$ and $\hat{p}(x)$ satisfy (up to numerical tolerances) the discrete counterpart of the steady-state PDEs. This is achieved by invoking MATLAB’s root-finding tools (i.e. `fsolve`) which act on the nonlinear system $\mathbf{f}(\mathbf{u}) = \mathbf{0}$, defined in (76), with one minor alteration. An additional integral constraint, namely $\int_0^1 P dx = 1$, is imposed to ensure that the overall number of anion vacancies is maintained within the perovskite. In order to preserve the system as a square system, we therefore replace the last anion vacancy conservation equation (53) in the nonlinear system $\mathbf{f}(\mathbf{u}) = \mathbf{0}$ with the equation

$$\sum_{i=0}^{i=N-1} \mathfrak{F}_{i+1/2}(P)A_{i+1/2} = 1. \tag{81}$$

We note that in the original PDE setting, the electric potential, ϕ , satisfies an elliptic equation, namely Poisson’s equation, and therefore does not require an initial condition, since the initial potential is uniquely defined by the initial densities. After spatial discretisation, Poisson’s equation is translated into a set of algebraic equations, see (54), (60) and (63), and as such, one would not expect initial conditions on the electric potential to be required. However, it is a requirement of `ode15s` that initial conditions for all components of a DAE system are specified. It is this feature of the integrator that requires us to find and specify the initial potential profile $\hat{\phi}(x)$; this profile is chosen to satisfy the algebraic equations derived from the discretisation of Poisson’s equation, and its boundary conditions, into which the initial conditions for P , n and p have been substituted.

5.3 Calculation of current densities and output

In order to calculate the total (dimensionless) current density $J(t)$ from the dimensionless solution generated by `IonMonger`, from (28), we use

$$J(t) = j^n_{k+1/2} + j^p_{k+1/2} - \frac{\epsilon_A V_T}{\hat{i}qG_0b^2\tau_{\text{ion}}} \left(\frac{\phi_{k+1} - \phi_k}{\Delta_{k+1/2}} \Big|_t - \frac{\phi_{k+1} - \phi_k}{\Delta_{k+1/2}} \Big|_{t-\hat{t}} \right) + \frac{D_1 \hat{N}_0}{G_0 b^2} F^P_{k+1/2}, \tag{82}$$

where $k = \text{ceil}\left(\frac{N+1}{2}\right)$ is the index of the grid point at (or nearest to) the midpoint of the perovskite layer and \hat{t} denotes the time since the previous time point. The quantities denoted by $j^n_{k+1/2}$, $j^p_{k+1/2}$ and $F^P_{k+1/2}$ are defined in (47), (48) and (46), respectively. Note that the contributions from the third and fourth terms are usually negligibly small in comparison with the sum of the first two terms.

In addition to the total current density, the (dimensionless) current density losses due to interface recombination (J_l and J_r) are calculated prior to re-dimensionalisation as follows:

$$J_l(t) = -\bar{R}_l \left(\frac{n_1}{k_E}, p_1 \right), \tag{83}$$

$$J_r(t) = -\bar{R}_r \left(n_N, \frac{p_N}{k_H} \right). \tag{84}$$

The final steps of the solution procedure carried out by `IonMonger` are to re-dimensionalise (see Sect. 3.5) and

then save the solution variables (\mathbf{P} , Φ , \mathbf{n} , \mathbf{p} , Φ^E , \mathbf{n}^E , Φ^H and \mathbf{p}^H), spatial vectors (\mathbf{x} , \mathbf{x}_E and \mathbf{x}_H), time (\mathbf{time}), evolution of the applied voltage (V) and evolution of the current densities defined above (\mathbf{J} , \mathbf{J}_I and \mathbf{J}_r). These data are saved into a `.mat` data file along with the input structure (`params`).

The saved data can be further analysed and plotted in MATLAB in any way chosen by the user. One example plotting script is included in the `IonMonger` GitHub repository, and instructions for its use are given in the `GUIDE`. This script can be used to plot the current density generated by a PSC during the reverse and forward scans of a J – V curve, alongside the current losses due to recombination at each of the perovskite/transport layer interfaces, as shown in Fig. 1a. Such a plot can enable the user to identify the limiting recombination mechanism for a particular set of input parameters as described in Sect. 2.

6 Validation

For the purpose of verifying the results generated by `IonMonger`, in Fig. 3a, we plot a measure of the error in eight different solution variables against the number of grid points ($N + N_E + N_H + 1$) on a logarithmic scale for five example simulations. The only input parameter that is varied between the five simulations is N , which takes values of 100, 200, 400, 800 and 1600, while all other input parameters are the same as those used for Fig. 1. Due to the lack of an exact solution, the errors are calculated with respect to another simulation performed on an even finer spatial grid consisting of 5613 points (for which the input parameter N is set equal to 3200). The chosen error measure is the sum (an l_1 norm) of the differences between the value of the variable computed by the example simulation and that computed by the 5613-point simulation, averaged over all 300 time points of the simulation protocol after $t = 0$. The same error measure was used in [6] to compare the same solution method applied to a single-layer version of the model against two other methods on different spatial grids. Here, the eight solution variables for which the error is calculated are the (dimensionless) ion vacancy density, electric potential, electron concentration and hole concentration at the ETL/perovskite and perovskite/HTL interfaces located at $x = 0$ and $x = 1$, respectively, as listed in the legend. The results demonstrate the expected second-order pointwise convergence of the finite element scheme on which `IonMonger` is based [6]. The variation in the magnitude of the error between the eight solution variables is due to differences in the magnitude of the dimensionless variables themselves.

Figure 3b shows the computation times associated with each of the five simulations in panel (a), also plotted against

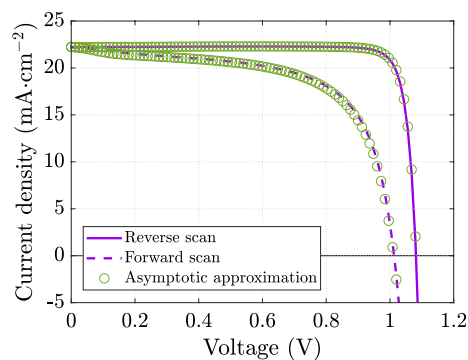


Fig. 4 A comparison between the current density calculated using the `IonMonger` code (purple lines) and an alternative, combined asymptotic/numerical method from [4, 5] (green circles) for the same simulation of a J – V measurement as shown in Fig. 1 (Color figure online)

the number of grid points on a logarithmic scale. Note that the computation time will also depend upon the length of the simulation protocol. The temporal tolerances for the integration in time performed by MATLAB's `ode15s` were fixed for all simulations at values of 10^{-6} for the relative tolerance and 10^{-10} for the absolute tolerance.

A comprehensive verification of the single-layer finite element scheme, from which this code was developed, is provided in Sections 5 and 6 of [6]. This work includes plots of each solution variable across the perovskite layer against corresponding asymptotic results, which show very good agreement between the two approaches for realistic values of the input parameters.

6.1 Comparison to asymptotic results

Further validation of the numerical scheme, against results obtained using an alternative (although approximate) solution method, is given in Fig. 4. Here, we compare the simulation results for the typical J – V measurement displayed in Fig. 1 to the equivalent quantities computed using a combined asymptotic/numerical method. This alternative method is described in detail for a single-layer model in [5] and has been used to explain trends in experimental observations in [4]. Excellent agreement is shown between the current densities computed using the two methods.

7 Conclusions

We have built a fast and robust numerical solver for coupled ionic–electronic charge transport in a realistic three-layer perovskite solar cell architecture. The scheme is able to simulate a variety of relevant device operating regimes, including current–voltage sweeps and open-circuit transients, both

with the possibility of having time-dependent illumination. Simulations of this sort can be carried out in seconds to minutes of computation time on a standard modern personal computer. The only prerequisite for making use of this tool is access to MATLAB and its suite of routines for time integration of ordinary differential equations, specifically the `ode15s` routine.

This work therefore provides a tool that is capable of playing a major role in guiding the development of perovskite solar cells. Our `IonMonger` code provides the possibility of independently varying each of the device parameters, so that their roles in determining cell performance can be discerned, something that is difficult, or even impossible, to achieve experimentally. One area of particular practical interest is understanding what can be done to mitigate the amount of parasitic recombination in PSCs, thereby further improving their performance. As demonstrated in [4], it is possible to suppress these losses via careful tuning of the cell's constituent material properties, and we anticipate that further studies in the same spirit will be made possible using the computational tool provided here. A second area where such a simulation tool is surely needed is in understanding the long-term degradation processes that occur within PSCs on timescales of between hours and weeks. While the current version of the code cannot simulate degradation, it can be used to predict the effects of different device parameters on some of the proposed causes of degradation. For example, degradation due to chemical reactions at the perovskite/transport layer interfaces [3] is likely to be exacerbated by iodide ion accumulation in the Debye layers, while degradation due to the penetration of extrinsic ions, e.g. oxygen, into the perovskite may be enhanced by an accumulation of ion vacancies [1]. The development of `IonMonger` to include additional physical processes that occur on longer timescales would allow researchers to investigate long-term behaviour and stability much more quickly than is possible via experimentation, and hence it will be the subject of future work.

The authors are committed to maintaining and expanding the functionality of the code, and any updates will be hosted on the GitHub repository which can be accessed at <https://github.com/PerovskiteSCModelling/IonMonger>. While we cannot promise a high level of technical support to all users, we are happy to receive any suggestions on ways in which the features of the code can be improved and/or expanded, and it is our intention that the code will grow as the research priorities of the perovskite community evolve. Contact details for current code developers can be found in the README file in the main folder of the `IonMonger` GitHub repository.

Acknowledgements NEC and JMC were supported by EPSRC-funded studentships from the CDT in New and Sustainable Photovoltaics (with reference EP/L01551X/1). NEC was subsequently supported by an EPSRC Doctoral Prize (Ref. EP/R513325/1). NEC would like to thank

Dr. Sam Mangham from the University of Southampton's Research Software Group for his advice on sustainable code design.

Open Access This article is distributed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits unrestricted use, distribution, and reproduction in any medium, provided you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license, and indicate if changes were made.

References

1. Aristidou, N., Eames, C., Sanchez-Molina, I., Bu, X., Kosco, J., Islam, M., Haque, S.A.: Fast oxygen diffusion and iodide defects mediate oxygen-induced degradation of perovskite solar cells. *Nat. Commun.* **8**, 15218 (2017)
2. Calado, P., Telford, A.M., Bryant, D., Li, X., Nelson, J., O'Regan, B.C., Barnes, P.R.F.: Evidence for ion migration in hybrid perovskite solar cells with minimal hysteresis. *Nat. Commun.* **7**, 13831 (2016)
3. Carrillo, J., Guerrero, A., Rahimnejad, S., Almora, O., Zarazua, I., Mas-Marza, E., Bisquert, J., Garcia-Belmonte, G.: Ionic reactivity at contacts and aging of methylammonium lead triiodide perovskite solar cells. *Adv. Energy Mater.* (2016). <https://doi.org/10.1002/aenm.201502246>
4. Courtier, N.E., Cave, J.M., Foster, J.M., Walker, A.B., Richardson, G.: How transport layer properties affect perovskite solar cell performance: insights from a coupled charge transport/ion migration model. *Energy Environ. Sci. Adv.* **12**(1), 396–409 (2019)
5. Courtier, N.E., Foster, J.M., O'Kane, S.E.J., Walker, A.B., Richardson, G.: Systematic derivation of a surface polarisation model for planar perovskite solar cells. *Eur. J. Appl. Math.* **30**, 427–457 (2019)
6. Courtier, N.E., Richardson, G., Foster, J.M.: A fast and robust numerical scheme for solving models of charge carrier transport and ion vacancy motion in perovskite solar cells. *Appl. Math. Model.* **63**, 329–348 (2018)
7. de Quillettes, D.W., Zhang, W., Burlakov, V.M., Graham, D.J., Leijtens, T., Oshero, A., Bulović, V., Snaith, H.J., Ginger, D.S., Stranks, S.D.: Photo-induced halide redistribution in organic-inorganic perovskite films. *Nat. Commun.* **7**, 11683 (2016)
8. Eames, C., Frost, J.M., Barnes, P.R.F., O'Regan, B.C., Walsh, A., Islam, M.S.: Ionic transport in hybrid lead iodide perovskite solar cells. *Nat. Commun.* **6**, 7497 (2015)
9. Egger, D.A., Bera, A., Cahen, D., Hodes, G., Kirchartz, T., Kronik, L., Lovrincic, R., Rappe, A.M., Reichman, D.R., Yaffe, O.: What remains unexplained about the properties of halide perovskites? *Adv. Mater.* **30**, 1–11 (2018)
10. Fell, A., Schön, J., Schubert, M.C., Glunz, S.W.: The concept of skins for silicon solar cell modeling. *Sol. Energy Mater. Sol. Cells* **173**, 128–133 (2017)
11. Foster, J.M., Snaith, H.J., Leijtens, T., Richardson, G.: A model for the operation of perovskite based hybrid solar cells: formulation, analysis, and comparison to experiment. *SIAM J. Appl. Math.* **74**, 1935–1966 (2014)
12. Idígoras, J., Contreras-Bernal, L., Cave, J.M., Courtier, N.E., Baranco, A., Borrás, A., Sánchez-Valencia, J.R., Anta, J.A., Walker, A.B.: The role of surface recombination on the performance of perovskite solar cells: effect of morphology and crystalline phase of TiO₂ contact. *Adv. Mater. Interfaces* **5**(21), 1801076 (2018). <https://doi.org/10.1002/admi.201801076>
13. Jacobs, D.A., Shen, H., Pfeffer, F., Peng, J., White, T.P., Beck, F.J., Catchpole, K.R.: The two faces of capacitance: new interpretations for electrical impedance measurements of perovskite solar

- cells and their relation to hysteresis. *J. Appl. Phys.* **124**, 225702 (2018)
14. MATLAB version 9.3.0.713579 (R2017b) (2017)
 15. National Renewable Energy Laboratory (NREL) research cell record efficiency chart. <http://www.nrel.gov/pv/assets/images/efficiency-chart.png> (2018). Accessed 17 June 2019
 16. Nelson, J.: *The Physics of Solar Cells*. Imperial College Press, London (2003)
 17. Phung, N., Abate, A.: The impact of nano- and microstructure on the stability of perovskite solar cells. *Small* **14**, 1–11 (2018)
 18. Richardson, G., O’Kane, S.E.J., Niemann, R.G., Peltola, T.A., Foster, J.M., Cameron, P.J., Walker, A.B.: Can slow-moving ions explain hysteresis in the current–voltage curves of perovskite solar cells? *Energy Environ. Sci.* **9**, 1476–1485 (2016)
 19. Scharfetter, D.L., Gummel, H.K.: Large-signal analysis of a silicon read diode oscillator. *IEEE Trans. Electron. Devices* **16**, 64–77 (1969)
 20. Shampine, L.F., Reichelt, M.W.: The MATLAB ODE suite. *SIAM J. Sci. Comput.* **18**, 1–22 (1997)
 21. Shampine, L.F., Reichelt, M.W., Kierzenka, J.A.: Solving index-1 DAEs in MATLAB and Simulink. *SIAM Rev.* **41**, 538–552 (1999)
 22. Skeel, R.D., Berzins, M.: A method for the spatial discretization of parabolic equations in one space variable. *SIAM J. Sci. Stat. Comput.* **11**, 1–32 (1990)
 23. Snaith, H.J.: Present status and future prospects of perovskite photovoltaics. *Nat. Mater.* **17**, 372–376 (2018)
 24. Snaith, H.J., Abate, A., Ball, J.M., Eperon, G.E., Leijtens, T.C., Noel, N.K., Stranks, S.D., Wang, J.T.-W., Wojciechowski, K., Zhang, W.: Anomalous hysteresis in perovskite solar cells. *J. Phys. Chem. Lett.* **5**, 1511–1515 (2014)
 25. van Reenen, S., Kemerink, M., Snaith, H.J.: Modeling anomalous hysteresis in perovskite solar cells. *J. Phys. Chem. Lett.* **6**, 3808–3814 (2015)
 26. Walsh, A., Scanlon, D.O., Chen, S., Gong, X.G., Wei, S.-H.: Self-regulation mechanism for charged point defects in hybrid halide perovskites. *Angew. Chem. Int. Ed.* **54**, 1791–1794 (2015)
 27. Walter, D., Fell, A., Wu, Y., Duong, T., Barugkin, C., Wu, N., White, T.P., Weber, K.: Transient photovoltage in perovskite solar cells: interaction of trap-mediated recombination and migration of multiple ionic species. *J. Phys. Chem. C* **122**, 11270–11281 (2018)

Publisher’s Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.