6-23-2023

A Novel Method for Alleviating Numerical Stiffness in Li-ion Thermal Abuse Models

Jason Ostanek
Purdue University

Mohammad Parhizi
UL Research Institutes

Judith Jeevarajan
UL Research Institutes

Recommended Citation

Ostanek, Jason & Parhizi, Mohammad & Jeevarajan, J.. (2023). A novel method for alleviating numerical stiffness in Li-ion thermal abuse models. Journal of Power Sources Advances. 23. 100123. 10.1016/j.powera.2023.100123.

Contents lists available at ScienceDirect

Journal of Power Sources Advances

journal homepage: www.sciencedirect.com/journal/journal-of-power-sources-advances

A novel method for alleviating numerical stiffness in Li-ion thermal abuse models

Jason Ostanek [a,*], Mohammad Parhizi [b], Judith Jeevarajan [b]

[a] School of Engineering Technology, Purdue University, 401 N. Grant St., West Lafayette, IN, 47907, USA
[b] Electrochemical Safety Research Institute, UL Research Institutes, 5000 Gulf Fwy, UHTB, Houston, TX, 77204, USA

ARTICLE INFO

Keywords:
Li-ion cells
Numerical simulation
Thermal runaway

ABSTRACT

Numerical modeling of thermal runaway in Lithium-ion batteries has become a critical tool for designing safer battery systems. Significant progress has been made in developing kinetic mechanisms for decomposition reactions and including additional physics such as venting and combustion. However, the governing heat conduction equation and decomposition reaction equations become numerically stiff during thermal runaway, which limits the utility of thermal abuse models to low-dimensional formulations. The present work introduces a new solution strategy, which switches from the full, 3D transient heat conduction formulation to an adiabatic, 0D lumped body formulation only during the stiff portion of the simulation, i.e., only during thermal runaway. To test the new solver, a 3D thermal abuse model was configured to simulate an oven test of an 18650-format cell. The new solver was exercised for scenarios of varying degrees of stiffness, and the results were compared with a baseline solver using typical integration methods. For an extremely stiff scenario, computation speed was increased by a factor of 183x relative to the baseline solver, with little impact on solution accuracy, thus effectively alleviating the numerical stiffness issue. The new solution strategy addresses the poor scalability of high-dimensional models, such as 3D-CFD-based thermal abuse models, and improves their practicality for industrial use.

1. Introduction

The number and scope of publications focusing on numerical modeling of thermal runaway in Lithium-ion batteries have grown steadily in recent years. In the *Journal of Power Sources*, for example, a search for “Lithium-ion thermal abuse model” revealed 129 publications in the 1990’s, 270 publications in the 2000’s, and 644 publications in the 2010’s. Thermal abuse models are now commonly used to conduct trade studies [1,2], diagnose faults [3,4], and design modules, which are more robust to propagating failure [5]. Kurzawski et al. [6], for example, used thermal abuse modeling in a trade study of spacing between cells and interstitial material in the design of a propagation resistant, pouch cell module.

Many thermal abuse and thermal propagation models are based on the principles of transient heat conduction [7–9]. The internal heat generation within the cell due to exothermic decomposition reactions is modeled using Arrhenius-type kinetics, and heat is conducted from the failing cell to its neighbors. Under certain conditions, conduction-based thermal abuse models are appropriate and perform well, viz. Kurzawski et al. [6]. However, capturing the effects of fire and/or ejecta becomes critically important in other scenarios. One clear example is shipping a group of low capacity cylindrical cells in a cardboard box. Flames emerging from a single cell or multiple cells may ignite the cardboard container, which results in additional energy release and heat transfer to adjacent cells [10]. The propagation rate is accelerated by the heat released from gas-phase and surface reactions, in addition to the heat conducted away from the failing cells.

Thermal abuse models that include venting and the subsequent heat transfer effects are necessary to improve the predictive capability in assessing the risk of propagation. The directionality of vent gases, flames, and ejecta may be captured using 3D-CFD models [11]. However, high-dimensional models which simulate the full thermal runaway process suffer from the large numerical stiffness arising from the nonlinear source terms in the energy equation. Unlike low-dimensional models, higher-dimensional models become prohibitive due to the combination of high numerical stiffness, which forces the solver to take small time steps, and large computational domains, which require a large number of mesh elements.

[*] Corresponding author.
E-mail address: jostanek@purdue.edu (J. Ostanek).

https://doi.org/10.1016/j.powera.2023.100123
Received 24 February 2023; Received in revised form 7 June 2023; Accepted 13 June 2023
Available online 23 June 2023

Despite this challenge, little work has been done to address the computational efficiency of thermal abuse models subject to high numerical stiffness. The present work addresses the high numerical stiffness problem using a novel approach, which takes advantage of the extremely high internal heating rate during thermal runaway. This allows the 3D formulation of the energy equation to be reduced to an adiabatic, 0D lumped body formulation only when numerical stiffness is highest, i.e., during thermal runaway. The reaction rate and conservation of energy equations are then integrated as a system of coupled ODEs during thermal runaway, when numerical stiffness is highest. The remainder of the paper describes the formulation of the model and new solution strategy, followed by a comparison of speed and accuracy for the new and baseline solvers applied to three different test cases having varying degrees of numerical stiffness.

2. Methods

2.1. 3-D thermal runaway model

The present 3D thermal abuse model formulation follows the approach presented by Kim et al. [12]. For a motionless and incompressible body, the conservation of energy equation reduces to the transient heat conduction equation [13]. Written in cylindrical coordinates, the transient heat conduction equation is the primary governing equation:

$$\rho C_p \frac{\partial T}{\partial t} = \frac{1}{r}\frac{\partial}{\partial r}\left(r \bullet k_r \frac{\partial T}{\partial r}\right) + \frac{1}{r^2}\frac{\partial}{\partial \theta}\left(k_\theta \frac{\partial T}{\partial \theta}\right) + \frac{\partial}{\partial z}\left(k_z \frac{\partial T}{\partial z}\right) + \sum \dot{Q}'''_{int,i} \tag{1}$$

where ρ and C_p are the density and specific heat capacity of the Li-ion cell, respectively, k_r is the radial thermal conductivity, k_θ is the circumferential thermal conductivity, k_z is the axial thermal conductivity, and $\dot{Q}'''_{int,i}$ is the volumetric heat generation rate due to the ith internal decomposition reaction. For each decomposition reaction, the volumetric heat generation rate is:

$$\dot{Q}'''_{int,i} = \frac{(H_i \bullet m_i \bullet R_i)}{V_{cell}} \tag{2}$$

where H_i is the heat of reaction, m_i is the initial mass of reactants, R_i is the reaction rate, and V_{cell} is the volume of the battery cell [12]. The reaction rates are usually described by Arrhenius-type expressions, such as the first-order reaction shown below:

$$-R_i = \frac{dx_i}{dt} = -A_i \exp\left(-\frac{E_{a,i}}{k_b \bullet T}\right) x_i \tag{3}$$

where x_i may represent a non-dimensional species concentration, non-dimensional length scale, or fractional degree of conversion, A_i is the frequency factor, $E_{a,i}$ is the activation energy, k_b is the Boltzmann constant, and T is temperature. In equations (1)–(3), the subscript i denotes a given decomposition reaction [12]. The present work considers four standard decomposition reactions: decomposition of the solid-electrolyte-interphase (SEI) layer [14,15], reaction of anode and electrolyte [14,15], reaction of cathode and electrolyte [14,16], and decomposition of electrolyte [12]. The reaction mechanisms are shown in Table 1. The kinetic parameters and initial conditions are listed in the Supplementary Material.

The governing partial differential equation (1) is subject to initial condition:

$$T(r, \theta, z, t=0) = T_0 \tag{4}$$

a mixed convection/radiation boundary condition at each outer surface:

$$-k_r \frac{\partial T}{\partial r}\bigg|_{r=R_o} = h(T - T_\infty) + \varepsilon\sigma\left(T^4 - T^4_\infty\right) \tag{5a}$$

Table 1
Reaction rate and source terms for the decomposition reactions.

Reaction Variable	Reaction Rate	Source Term	Reference
Dimensionless Amount of lithium-containing meta-stable species in the SEI layer, x_{sei}	$\frac{dx_{sei}}{dt} = -k_{sei}$	$k_{sei} = A_{sei} \exp\left(-\frac{E_{a,sei}}{k_b T}\right) x_{sei}$	[14,15]
Dimensionless amount of lithium in the anode, x_{AnE}	$\frac{dx_{AnE}}{dt} = -k_{AnE}$	$k_{AnE} = A_{AnE} \exp\left(-\frac{E_{a,AnE}}{k_b T}\right) \exp\left(-\frac{z}{z_0}\right) x_{AnE}$	[14,15]
Dimensionless SEI thickness, z	$\frac{dz}{dt} = k_{tSEI}$	$k_{tSEI} = -k_{AnE}$	[14,15]
Cathode-electrolyte reaction degree of conversion, α_{cat}	$\frac{d\alpha_{cat}}{dt} = k_{cat}$	$k_{cat} = A_{cat} \exp\left(-\frac{E_{a,cat}}{k_b T}\right)(\alpha_{cat})(1-\alpha_{cat})$	[14,16]
Dimensionless electrolyte concentration, x_{elec}	$\frac{dx_{elec}}{dt} = -k_{elec}$	$k_{elec} = A_{elec} \exp\left(-\frac{E_{a,elec}}{k_b T}\right)(x_{elec})$	[12]

and an adiabatic boundary condition at the top and bottom surfaces:

$$-k_z \frac{\partial T}{\partial z}\bigg|_{z=0} = 0 \tag{5b}$$

$$-k_z \frac{\partial T}{\partial z}\bigg|_{z=L} = 0 \tag{5c}$$

In equation (5), R_o is the outer radius of the cell, L is the length of the cell, h is the convection coefficient, T_∞ is the freestream temperature, and ε is the surface emissivity. The reaction rate ordinary differential equation (3) is subject to initial condition:

$$x_i(r, \theta, z, t=0) = x_0 \tag{6}$$

All model input parameters needed for equation (1) through (6) are reported in the Supplementary Material.

2.2. Baseline solution strategy

The baseline solver is implemented using a commercial, finite volume solver to discretize and integrate the governing equations [17,18]. The computational domain comprised a single cylinder with an 18 mm diameter and 65 mm height, representing an 18650 format Li-ion cell. The computational mesh uses 26,250 structured, hexahedral cells, as shown in Fig. 1(a), and a schematic of the baseline solver algorithm is shown in Fig. 1(b).

The solution is initialized, and the solver enters the outer, time-stepping loop at timestep $N = 1$. The time step size is adjusted during the simulation using an adaptive time-stepping algorithm, following Parhizi et al. [19], where the associated parameters are reported in the Supplementary Material. The solver then enters the inner loop, at iteration $n = 1$. During the first iteration of the inner loop, the reaction rates are updated, the volumetric heat source term is updated, and the reaction rates are integrated using the fourth-order Runge-Kutta method. One advantage of using explicit Runge-Kutta methods to integrate the decomposition reaction equations is avoiding the re-computation of the heat source term during each iteration of the inner loop, as opposed to implicit methods [19]. The transient heat conduction equation (1) is a partial differential equation integrated using second-order implicit temporal discretization and second-order central difference spatial discretization. The solver checks for convergence of equation (1) and increases the iteration number of the inner loop, if necessary. Otherwise, the solver proceeds to the next time step and continues until the simulation time exceeds the final time, t_{end}.

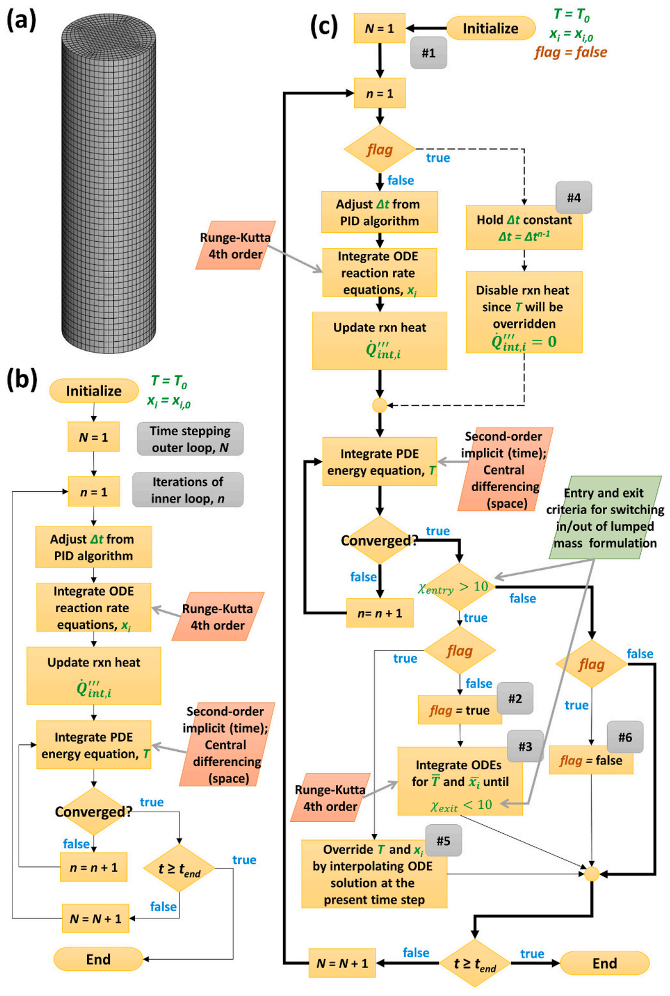
(a)
(b)
Initialize
T = T_0
x_i = x_i,0
N = 1
Time stepping outer loop, N
n = 1
Iterations of inner loop, n
Adjust Δt from PID algorithm
Integrate ODE reaction rate equations, x_i
Runge-Kutta 4th order
Update rxn heat Q̇'''int,i
Integrate PDE energy equation, T
Second-order implicit (time); Central differencing (space)
Converged?
true
false
n = n + 1
t ≥ t_end
true
false
N = N + 1
End
(c)
N = 1
Initialize
T = T_0
x_i = x_i,0
flag = false
#1
n = 1
flag
true
false
Adjust Δt from PID algorithm
#4
Hold Δt constant
Δt = Δt^(n-1)
Runge-Kutta 4th order
Integrate ODE reaction rate equations, x_i
Disable rxn heat since T will be overridden
Q̇'''int,i = 0
Update rxn heat Q̇'''int,i
Integrate PDE energy equation, T
Second-order implicit (time); Central differencing (space)
Entry and exit criteria for switching in/out of lumped mass formulation
Converged?
true
false
χentry > 10
false
true
n = n + 1
flag
true
false
#2
flag = true
flag
false
true
#6
flag = false
#3
Integrate ODEs for T̄ and x̄_i until χexit < 10
Runge-Kutta 4th order
#5
Override T and x_i by interpolating ODE solution at the present time step
N = N + 1
false
t ≥ t_end
true
End

Fig. 1. Computational domain using structured, hexahedral cells (a), schematic of baseline thermal abuse solver (b), and schematic of new thermal abuse solver (c).

2.3. New solution strategy

The new solver uses the same discretization schemes, adaptive time-stepping algorithm, and overall logic as the baseline solver. The primary difference compared with the baseline model is the treatment of the rapid heating period that takes place during thermal runaway. At a certain time during thermal runaway, the rate of internal heat generation far exceeds the rate of heat conduction within the cell and external heat dissipation at the cell boundaries. Under this condition, the 3D transient heat conduction equation (1) may be reduced to an adiabatic, 0D lumped body formulation. The resulting ordinary differential equation (ODE) is:

$$mC_p\frac{d\overline{T}}{dt} = \sum \dot{Q}_{int,i} \tag{7}$$

where *m* is the mass of the cell, $\overline{T}$ is the volume-averaged temperature of the battery cell, and the reaction heat source terms, $\dot{Q}_{int,i}$, are total heat generation rate rather than the volumetric heat generation rates in equation (1).

The criterion for switching from the full 3D transient heat conduction equation (1) to lumped mass formulation (7) was based on the ratio of total internal heat generation rate to the internal heat conduction rate:

$$\chi_{entry} = \frac{\sum \dot{Q}^{'''}_{int,i}}{\left|\frac{1}{r}\frac{\partial}{\partial r}\left(r \bullet k_r\frac{\partial T}{\partial r}\right) + \frac{1}{r^2}\frac{\partial}{\partial \theta}\left(k_\theta\frac{\partial T}{\partial \theta}\right) + \frac{\partial}{\partial z}\left(k_z\frac{\partial T}{\partial z}\right)\right|} \tag{8}$$

The model formulation was switched to equation (7) when χ_{entry} exceeded a value of 10 at all points in the volume of the battery cell. Once the solver enters lumped mass formulation, equation (7), there will be no internal temperature gradient and χ_{entry} becomes infinite. To define a suitable exit criterion, the ratio of internal heat generation rate to convective and radiative heat dissipation rate at the surface of the cell was calculated:

$$\chi_{exit} = \frac{\sum \dot{Q}^{'''}_{int,i} V_{cell}}{\left|hA_s(\overline{T} - T_\infty) + \varepsilon A_s\sigma\left(\overline{T}^4 - T_\infty^4\right)\right|} \tag{9}$$

where A_s is the convective surface area (other parameters are defined in the Supplementary Material). The exit criterion was defined when the value of χ_{exit} decreased below a value of 10 at all points in the volume of the battery cell.

Since the governing equation (7) and reaction rate equation (3) are now a system of coupled ODEs, the integration with respect to time can be accomplished practically instantaneously, relative to the time required to integrate the full 3D transient heat conduction equation throughout thermal runaway. A schematic of the new solver is shown in Fig. 1(c), and the solution procedure is as follows.

1. Prior to thermal runaway, integrate equations (1) and (3) using the same methods as the baseline solver. Refer to the annotated (#1) on Fig. 1(c) and follow the bold lines to visualize the process flow of the baseline solver.
2. When the value of χ_{entry} exceeds 10 at all points in the jellyroll, pause the solver and calculate the volume-average temperature, $\overline{T}$, and the volume-average reaction variables, $\overline{x}_i$, for the most recent time step. A flag variable is used to track the state of the solver, initially set as *flag* = false. When the entry criterion is met, set *flag* = true, as shown by the annotated (#2) on Fig. 1(c).
3. Re-cast the transient heat conduction equation into an adiabatic, lumped body ODE, shown in equation (7). Integrate both (7) and (3) using the fourth-order Runge-Kutta method and adaptive time-stepping where the initial conditions are taken from the volume-averaged values calculated in step #2. Stop integration when the value of χ_{exit} decreases below a value of 10. After this integration is complete, the time history of temperature and reaction variables are saved in a data file. Refer to the annotated (#3) on Fig. 1(c).
4. Resume time-marching but disable the adaptive time-stepping algorithm such that the time step is now constant and held equal to the most recent time step size from step #2. The reaction heat source term, equation (2), is then artificially set to zero to reduce the stiffness of the energy equation. Refer to the annotated (#4) and follow the dashed lines on Fig. 1(c).
5. After convergence is reached in the inner loop, apply $\overline{T}$ and $\overline{x}_i$ uniformly throughout the cell volume by interpolating from the ODE solution data file that was stored in step #3. Refer to the annotated (#5) on Fig. 1(c).
6. When χ_{exit} decreases below 10, resume integration using the baseline solver until the final solution time is reached. Set *flag* = false to resume the normal solution procedure. Refer to the annotated (#6) on Fig. 1(c).

2.4. Reference model

A reference model was established using MATLAB Simulink for the purpose of quantifying the accuracy of the baseline and new solvers. As previously mentioned, the present work solves the 3D transient heat conduction equation to highlight the challenges of integrating the governing equation (1) and constituent equation (3) for a representative 3D mesh under stiff numerical conditions. Though the problem is set up using a 3D formulation and mesh, the boundary conditions (5) only allow heat transfer in the radial direction, making problem 1D and, therefore, suitable for development as a Simulink model. The 3D formulation is required for problems that are non-axisymmetric, such as: patch heater or other trigger methods the heat one part of the cell locally, module or pack geometry that is non-axisymmetric, venting and combustion that is non-axisymmetric, and other thermal boundary conditions (cooling) that are non-axisymmetric.

The primary differences between the reference model and the full 3D model lie in the spatial and temporal discretization. The spatial discretization scheme is the same as the baseline and new solvers, i.e., second order central difference, but the spatial resolution is increased by about 3.6x through the use of 90 concentric rings to capture heat transfer in the radial direction. Time discretization was accomplished using MATLAB's ODE15s solver, with a high relative tolerance of 1E-9. The high spatial resolution and high relative tolerance result in an accurate model for evaluating the baseline and new solvers.

3. Results and discussion

Three cases were considered to evaluate the performance of the new solver: mildly stiff, moderately stiff, and extremely stiff. The reaction heat of the electrolyte decomposition reaction, H_{elec}, was manipulated to determine the stiffness of the problem. The reaction heat increased from $H_{elec} = 155$ J/g to 775 J/g and further to 1550 J/g for the mild, moderate, and stiff cases, respectively. Simulations were conducted on a high-performance cluster and carried out in parallel on 48 cores for all models and cases.

Fig. 2(a) shows volume-averaged temperature *vs.* time for the baseline and new solvers, where the moderate and extreme cases are offset

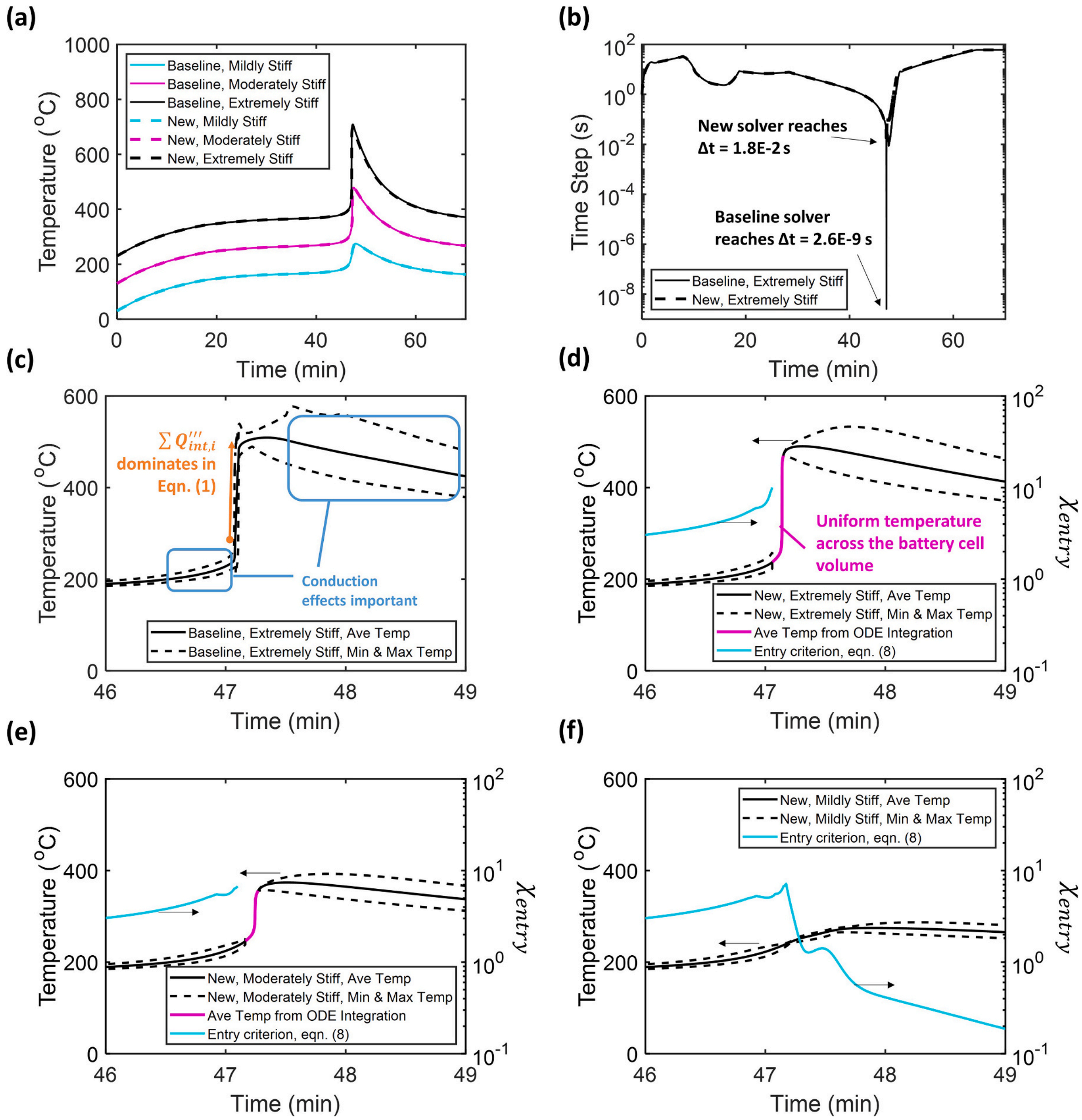

Fig. 2. Temperature *vs.* time of oven test simulation of an 18650 cell with comparison of baseline and new solvers under three cases of increasing numerical stiffness where each case is offset by 100 °C for clarity (a), time step size vs time comparison for the extremely stiff case (b), zoom-in of minimum, average, and maximum temperature *vs.* time for baseline solver during the extremely stiff case (c), and zoom-in of minimum, average, and maximum temperature *vs.* time with the entry criterion shown on the second y-axis for the extremely (d), moderately (e), and mildly stiff cases (f).

by +100 °C and +200 °C, respectively, for visualization purposes. In general, the baseline and new solvers agree well. Under all three cases, the cell goes into thermal runaway at t ≈ 47 min. The peak volume-averaged temperatures were 275 °C, 379 °C, and 509 °C for the baseline solver under mild, moderate, and extreme numerical stiffness, respectively. The new solver reached similar peak temperatures: 275 °C, 374 °C, and 490 °C. These results confirm that increasing electrolyte decomposition reaction effectively increases the numerical stiffness of the problem without affecting the initial heating period.

Fig. 2(b) shows the time step size *vs.* time for the baseline and new solvers during the extremely stiff case. As the cell enters thermal runaway (*t* ≈ 47 min), the baseline solver was forced to choose very small time step sizes, reaching a minimum value of 2.6E-9 s. The new solver, on the other hand, showed a minimum time step size of 1.3E-2 s. It should be noted that the integration of the coupled ODEs during thermal runaway in the new solver still requires small time step sizes. For the extremely stiff scenario, the coupled ODE solver (step #3 in Fig. 1) reached a minimum time step size of 1.7E-8 s. Despite the small time step requirement when integrating the coupled ODEs, the global time step size of the new solver was not reduced to small time steps since the solution during thermal runaway is calculated *a priori* and applied uniformly to the cell volume at each global time step.

Fig. 2(c) and (d) show the volume-average, minimum, and maximum temperature within the cell volume *vs.* time for the baseline and new solvers, respectively. The annotations in Fig. 2(c) illustrate that conduction effects are important in the moments leading up to thermal runaway when there is a noticeable temperature gradient within the cell. At the moment of thermal runaway, the internal heat generation rate exceeds the rate of heat conduction within the cell and external heat dissipation at the cell's boundary, signaled by the sharp increase in temperature. Conduction effects become important again after thermal runaway, during cooldown. The annotation in Fig. 2(d) provides a visualization for the new solver's treatment of the thermal runaway process. The temperature distribution within the cell collapses to a uniform temperature when the solver switches to the adiabatic, lumped body approximation. After the temperature rise, the solver switches back to the baseline solver, and the internal temperature distribution is recovered for the cooldown process.

Fig. 3(a) shows the computational error, calculated as the difference between the volume-averaged temperature of the baseline or new solvers relative to the reference model. Errors increase with numerical stiffness, and the largest errors occur at the moment of thermal runaway, which is caused by the difference in the onset time of thermal runaway. Aside from the large error spike caused by the slight difference in thermal runaway onset time, the baseline model showed an error of 16 °C at *t* = 48 min, which decreased to 5 °C at *t* = 60 min for the extremely stiff case. The new model showed an error of 5 °C at *t* = 49 min decreasing to 3 °C at *t* = 60 min. In addition, the new solver showed thermal runaway onset 7 s before the reference model, and the baseline model showed thermal runaway 10 s before the reference model. These results indicate that the accuracy of the new and baseline solvers was comparable.

Fig. 3(b) compares the simulation time for baseline and new solvers for all three cases. The poor scalability of the baseline solver is clearly seen as the wall-clock time increased from 10.9 min for the mildly stiff case, to 17.7 min for the moderately stiff case, and to 2416.3 min for the extremely stiff case. It is possible to increase the computation speed of the baseline solver by simply limiting the minimum time step size to 1E-6s, whereas all previous simulations had no minimum time step size limit. This modification has no effect on the mild and moderate cases, since the solver does not take time steps smaller than 1E-6 s. For the extremely stiff case, the wall clock time was reduced to 144.3 min, which is a 16.7x decrease in wall-clock time. The error relative to the reference model was comparable to the baseline (without time step size limit) and new solvers. The effectiveness of the new solver in alleviating numerical stiffness is clearly observed in Fig. 3(b). The wall-clock time for the extremely stiff case was 13.2 min, which is a 183x decrease in wall clock time from the baseline solver and a 11x decrease from the baseline solver with time step size limited to 1E-6 s.

It should be noted that the baseline and new solvers showed no practical issues with numerical stability for the problems considered. Although the non-linear heat source terms increase exponentially with increasing temperature, the consumption of reactants limits the heat release. While the solvers showed no stability issues, the time step should be carefully monitored to ensure accurate solution.

4. Conclusions

A new solution method was presented for thermal abuse simulations of Li-ion battery cells. The new solver reduces numerical stiffness commonly encountered during thermal runaway by simplifying the 3D transient heat conduction equation to an adiabatic, 0D lumped body formulation only during the numerically stiff portions of the thermal abuse simulation, i.e., during thermal runaway. This assumption is valid since the internal heat generation rate during thermal runaway is vastly greater than the rate of heat conduction within the cell and external heat transfer across the cell's boundary. The new solver exhibits excellent scaling where computational speed was increased by a factor of 183x for an extremely stiff scenario. The new solver showed similar levels of accuracy relative to the baseline solver using traditional integration methods.

The new solver improves the practicality of high-dimensional thermal abuse models with large mesh sizes, such as 3D, CFD-based models. The computational challenge with large mesh size is no longer compounded by the numerical stiffness arising in the energy equation and reaction rate equations. In moving towards highly resolved simulations

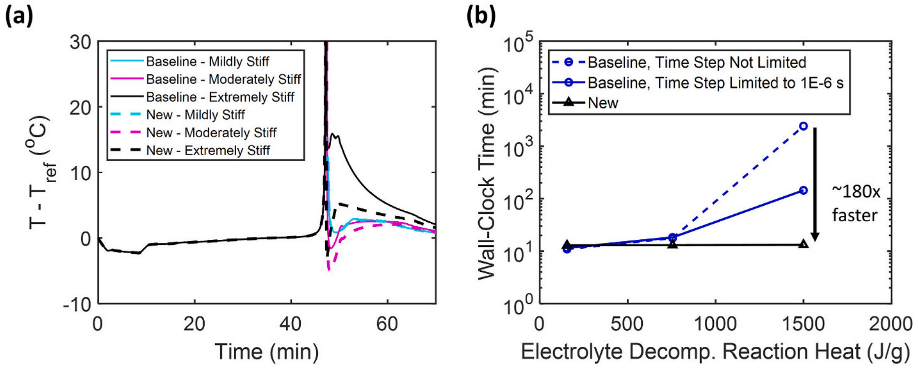

Fig. 3. Error of baseline and new solvers relative to the reference model (a) and comparison of wall-clock time *vs.* electrolyte decomposition reaction heat for: baseline solver with no limit on the minimum time step size, baseline solver with time step size limited to 1E-6 s, and new solver (c).

with advanced physics such as venting, combustion, and discrete phase particulates, improvements in the numerical treatment of the various aspects of these models, such as the novel approach for treating thermal runaway presented here, are critical.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

Acknowledgements

Research reported in this publication was supported by the Electrochemical Safety Research Institute, UL Research Institutes.

Appendix A. Supplementary data

Supplementary data to this article can be found online at https://doi.org/10.1016/j.powera.2023.100123.

References

[1] Q. Xia, et al., Safety risk assessment method for thermal abuse of lithium-ion battery pack based on multiphysics simulation and improved bisection method, Energy 264 (2023/02/01/2023), 126228, https://doi.org/10.1016/j.energy.2022.126228.

[2] Y. Takagishi, Y. Tozuka, T. Yamanaka, T. Yamaue, Heating simulation of a Li-ion battery cylindrical cell and module with consideration of gas ejection, Energy Rep. 8 (2022) 3176–3188, https://doi.org/10.1016/j.egyr.2022.02.086.

[3] S. Dey, et al., Model-based real-time thermal fault diagnosis of Lithium-ion batteries, Control Eng. Pract. 56 (2016/11/01/2016) 37–48, https://doi.org/10.1016/j.conengprac.2016.08.002.

[4] W. Huang, et al., Early warning of battery failure based on venting signal, J. Energy Storage 59 (2023/03/01/2023), 106536, https://doi.org/10.1016/j.est.2022.106536.

[5] Q. Li, et al., Numerical investigation of thermal runaway mitigation through a passive thermal management system, J. Power Sources 429 (2019/07/31/2019) 80–88, https://doi.org/10.1016/j.jpowsour.2019.04.091.

[6] A. Kurzawski, L. Torres-Castro, R. Shurtz, J. Lamb, J.C. Hewson, Predicting cell-to-cell failure propagation and limits of propagation in lithium-ion cell stacks, Proc. Combust. Inst. 38 (3) (2021/01/01/2021) 4737–4745, https://doi.org/10.1016/j.proci.2020.06.270.

[7] C.F. Lopez, J.A. Jeevarajan, P.P. Mukherjee, Characterization of lithium-ion battery thermal abuse behavior using experimental and computational analysis, J. Electrochem. Soc. 162 (10) (2015) A2163–A2173, https://doi.org/10.1149/2.0751510jes.

[8] P.T. Coman, E.C. Darcy, C.T. Veje, R.E. White, Numerical analysis of heat propagation in a battery pack using a novel technology for triggering thermal runaway, Appl. Energy 203 (Oct 1 2017) 189–200, https://doi.org/10.1016/j.apenergy.2017.06.033 (in English).

[9] X. Feng, L. Lu, M. Ouyang, J. Li, X. He, A 3D thermal runaway propagation model for a large format lithium ion battery module, Energy 115 (2016) 194–208, https://doi.org/10.1016/j.energy.2016.08.094.

[10] M. Chen, D. Ouyang, J. Liu, J. Wang, Data and video for the thermal and fire propagation of multiple lithium-ion batteries, Data Brief 26 (2019/10/01/2019), 104379, https://doi.org/10.1016/j.dib.2019.104379.

[11] W. Li, V. León Quiroga, K.R. Crompton, J.K. Ostanek, High resolution 3-D simulations of venting in 18650 lithium-ion cells (in English), Front. Energy Res. 9 (2021), https://doi.org/10.3389/fenrg.2021.788239. December-07 2021.

[12] G.-H. Kim, A. Pesaran, R. Spotnitz, A three-dimensional thermal abuse model for lithium-ion cells, J. Power Sources 170 (2) (2007) 476–489, https://doi.org/10.1016/j.jpowsour.2007.04.018.

[13] D. Poulikakos, Conduction Heat Transfer, Prentice-Hall, Englewood Cliffs, New Jersey, 1994.

[14] T.D. Hatchard, D.D. MacNeil, A. Basu, J.R. Dahn, Thermal model of cylindrical and prismatic lithium-ion cells, J. Electrochem. Soc. 148 (7) (2001) A755–A761, https://doi.org/10.1149/1.1377592.

[15] M.N. Richard, J.R. Dahn, Accelerating rate calorimetry study on the thermal stability of lithium intercalated graphite in electrolyte II. Modeling the results and predicting differential scanning calorimeter curves, J. Electrochem. Soc. 146 (6) (Jun 1999) 2078–2084, https://doi.org/10.1149/1.1391894 (in English).

[16] D.D. MacNeil, L. Christensen, J. Landucci, J.M. Paulsen, J.R. Dahn, An autocatalytic mechanism for the reaction of Li[sub x]CoO[sub 2] in electrolyte at elevated temperature, J. Electrochem. Soc. 147 (3) (2000) 970, https://doi.org/10.1149/1.1393299.

[17] Ansys, ANSYS FLUENT 2022/R1 user's guide. http://www.ansyhelp.ansys.com.

[18] Ansys, ANSYS FLUENT 2022/R1 theory guide. http://www.ansyhelp.ansys.com.

[19] M. Parhizi, A. Jain, G. Kilaz, J.K. Ostanek, Accelerating the numerical solution of thermal runaway in Li-ion batteries, J. Power Sources 538 (2022), https://doi.org/10.1016/j.jpowsour.2022.231531.