COMPUTATIONAL STUDY OF TWO-PHASE STEFAN PROBLEM IN A FINITE INTERVAL

Daniel Lee

Department of Applied Mathematics
Tunghai University
Taichung, 40704, TAIWAN, R.O.C.

Abstract: We consider in this work the heat diffusion of one-dimensional spatial variable in finite domains. Two finite interval models are derived from relevant models in the semi-infinite interval. We investigate both static and dynamic grid approaches in fixed coordinates. Numerical discretizations based on finite differences or spectral differentiations are applied. For each computational model we discuss the numerical methods and the issues in software design and the trade-offs between accuracy and efficiency. In particular, we discuss a control variable (the scale factor) for achieving appropriate grid distribution and therefore better accuracy and run-time efficiency. The observations can be helpful in practical applications for which no explicit solution is expected.

AMS Subject Classification: 65-05, 65M06

Key Words: heat transfer, Stefan problem, spline, collocation

Nomenclature

α – the Stefan constant

Δt – temporal meshsize

Δx – spatial meshsize

Received: April 16, 2013

© 2013 Academic Publications, Ltd.
url: www.acadpubl.eu

k_I – thermal conductivity of the solid phase (ice)
 k_W – thermal conductivity of the liquid phase (water)
 S – one-point interface as the water-ice boundary in the semi-infinite interval
 T_{Ice} – constant temperature at the far-field
 T_{Water} – constant temperature at the left boundary
 t – the temporal variable in the semi-infinite interval
 u – temperature distribution in the semi-infinite interval, $^{\circ}\text{C}$
 v – transformed image of u in the finite interval, $^{\circ}\text{C}$
 x – spatial coordinate in models in the semi-infinite interval
 ξ – spatial coordinate in models in the finite interval
 ξ_S – transformed image of S in the finite interval

Subscripts

t – temporal derivatives in the semi-infinite interval
 τ – temporal derivatives in the finite interval
 x – spatial derivatives in the semi-infinite interval
 ξ – spatial derivatives in the finite interval

Matrices

C – collocation matrix in water or ice zone subject to Dirichlet B.C.
 D – first-order differentiation matrix in water or ice zone
 $D2$ – second-order differentiation matrix in water or ice zone

Experiments

e_{av} – normalized 1-norm of errors in the calculated temperature over all spatial nodes
 e_{max} – maximum norm of errors in the calculated temperature over all spatial nodes
 e_S – signed error in the calculated interface position
 n_I – the number of computational nodes in the ice zone
 n_t – total number of time steps in marching
 n_W – the number of computational nodes in the water zone
 n_x – the number of interior computational nodes in the spatial variable
 r_{av} – normalized 1-norm of the residual vector in calculation of the discrete algebraic system
 r_{max} – maximum norm of the residual vector in calculation of the discrete algebraic system
 $t\text{-march}$ – the *cpu-time* in the whole time marching including intermediate outputs to screen
 $t\text{-nl-solve}$ – the *cpu-time* in solving the discrete system, with pre- and post- operations.

1. Introduction

We consider in this work the temperature function $u = u(x, t)$, defined for temporal variable $t \geq 0$ and one-dimensional spatial variable x in a finite interval. Traditionally the study of directional solidification was and is performed within the framework of the classical model [23], leading to the Stefan boundary-value problem [9, 26] in the semi-infinite interval. A more practical model was investigated in [1, 2], which considered a mushy layer in addition to solid and liquid phases. This mushy layer model was used in [13] to solve the real-life problem [16, 28], based on a pre-study [12] in the classical two-phase model in the semi-infinite interval. However, in the field of computational differential equations, many numerical methods are developed for applications in a finite interval. Hoping to take advantages of established knowledge in computation involving finite intervals, the author convert the semi-infinite interval model to a system in the finite interval $[-1, 1]$ and investigate the numerical modeling in this paper.

Two spatial transforms (mappings) are applied in Section 2 to yield models of Stefan problems in the finite interval $[-1, 1]$. Numerical methods for solving these models are investigated in Sections 3,4. Followed in Section 5 are the numerical experiments and discussions. Conclusion is given in the final section. More tables and figures appear in the appendices.

The author has been working on solving the real-life problem [16, 28], for which no explicit solution is expected and very long-term computations are needed. Many decisions in the associated DNS (Direct Numerical Simulation) are made based on observations noted from kind of burning tests in regards of the classical two-phase models. Issues of numerical modeling of the classical

Model S. Two-phase Stefan problem in $[0, +\infty)$ in fixed coordinate.

were addressed in some details in [12], which we refer to for explicit formulation, the analytic similarity solution, and for formulation in moving coordinate. Plots of the solution and its temporal derivative at several specific time instances and two-hundred spatial nodes are shown in Fig. 1. We mention that the phase field function approach ([9, 11]) is not considered in this work.

2. Stefan Problem in a Finite Interval

In the field of computational differential equations, many numerical methods are developed for applications in a finite interval. We therefore convert the semi-infinite interval model to a system in the finite interval $[-1, 1]$. Considered here are a coordinate transform and its inverse $\xi = \xi(x)$, $x = x(\xi)$, where x denotes spatial variable

Figure 1: Analytic solutions and temporal derivatives at eight equally-distanced times up to $t_{end} = 12$ in the semi-infinite interval truncated differently.

in the semi-infinite interval $[0, +\infty)$ and ξ in the finite interval $[-1, 1)$. The relation $v(\xi, t) = u(x, t)$ and application of chain rule yields

$$u_t = v_t, \quad u_x = \xi_x v_\xi, \quad u_{xx} = (\xi_x)^2 v_{\xi\xi} + \xi_{xx} v_\xi, \quad \xi'_S(t) = \xi_x S'(t),$$

where ξ_S denotes the interface coordinate in $[-1, 1)$ as a function of time.

Two such transforms will be discussed in the sequel.

2.1. Exponentially Transformed System in a Finite Interval

With a real positive scale factor x_{ref} , we consider the exponential transform and its inverse

$$\begin{aligned}\xi(x) &= 1 - 2e^{-\frac{\log 2}{x_{\text{ref}}}x} = 1 - 2^{1 - \frac{x}{x_{\text{ref}}}}, \\ x(\xi) &= x_{\text{ref}} - \frac{x_{\text{ref}}}{\log 2} \log(1 - \xi).\end{aligned}$$

This maps the triple $x \in \{0, x_{\text{ref}}, +\infty\}$ to $\xi \in \{-1, 0, 1\}$. For uniform nodes in the finite interval $[-1, 1]$ with mesh size $\varepsilon = \frac{2}{n}$, the right-most interior nodes ($\xi = 1 - \varepsilon$) corresponds to $x = \frac{\log n}{\log 2} x_{\text{ref}}$ in the semi-infinite interval.

We are thus led to the transformed problem.

Model F1. Exponentially mapped system onto the finite interval $[-1, 1]$.

Water zone, $-1 \leq \xi \leq \xi_S$,

$$\text{PDE : } v_t = k_W \left(\frac{\log 2}{x_{\text{ref}}} \right)^2 [(1 - \xi)^2 v_{\xi\xi} - (1 - \xi) v_\xi], \quad -1 < \xi < \xi_S, \quad 0 < t. \quad (2.1)$$

$$\begin{aligned}\text{B.C. : } v(\xi = -1, t) &= T_{\text{Water}} > 0, & 0 \leq t, \\ v(\xi = \xi_S, t) &= 0, & 0 \leq t.\end{aligned}$$

Interface, $\xi = \xi_S$,

$$\text{ODE : } \alpha \frac{d\xi_S(t)}{dt} = \left(\frac{\log 2}{x_{\text{ref}}} \right)^2 (1 - \xi_S)^2 \left[k_I v_\xi(\xi_S^+) - k_W v_\xi(\xi_S^-) \right]. \quad (2.2)$$

$$\text{I.C. : } \xi_S(t = 0) = -1.$$

Ice zone, $\xi_S \leq \xi < 1$,

$$\text{PDE : } v_t = k_I \left(\frac{\log 2}{x_{\text{ref}}} \right)^2 [(1 - \xi)^2 v_{\xi\xi} - (1 - \xi) v_\xi], \quad \xi_S < \xi < 1, \quad 0 < t. \quad (2.3)$$

$$\begin{aligned}\text{B.C. : } v(\xi = \xi_S, t) &= 0, & 0 \leq t, \\ v(\xi = 1^-, t) &= T_{\text{Ice}} < 0, & 0 \leq t.\end{aligned}$$

2.2. Algebraically Transformed System in a Finite Interval

A second transform and its inverse are the following

$$\begin{aligned}\xi(x) &= 1 - \frac{2x_{\text{ref}}}{x+x_{\text{ref}}} = \frac{x-x_{\text{ref}}}{x+x_{\text{ref}}} = \frac{\frac{x}{x_{\text{ref}}}-1}{\frac{x}{x_{\text{ref}}}+1}, \\ x(\xi) &= x_{\text{ref}} \frac{1+\xi}{1-\xi}.\end{aligned}$$

This maps the triple $x \in \{0, x_{\text{ref}}, +\infty\}$ to $\xi \in \{-1, 0, 1\}$. We note for uniform nodes in the finite interval $[-1, 1]$ with mesh size $\varepsilon = \frac{2}{n}$, the right-most interior node ($\xi = 1 - \varepsilon$) corresponds to $x = (n - 1)x_{\text{ref}}$ in the semi-infinite interval.

We can derive and obtain the transformed problem.

Model F2. Algebraically mapped system onto the finite interval $[-1, 1]$.

Water zone, $-1 \leq \xi \leq \xi_S$,

$$\begin{aligned}\text{PDE : } v_t &= k_W \left[\frac{(1-\xi)^4}{4x_{\text{ref}}^2} v_{\xi\xi} - \frac{(1-\xi)^3}{2x_{\text{ref}}^2} v_{\xi} \right], \quad -1 < \xi < \xi_S, \quad 0 < t. \\ \text{B.C. : } v(\xi = -1, t) &= T_{\text{Water}} > 0, \quad 0 \leq t, \\ v(\xi = \xi_S, t) &= 0, \quad 0 \leq t.\end{aligned}\tag{2.4}$$

Interface, $\xi = \xi_S$,

$$\begin{aligned}\text{ODE : } \alpha \frac{d\xi_S(t)}{dt} &= \frac{(1-\xi_S)^4}{4x_{\text{ref}}^2} [k_I v_{\xi}(\xi_S^+) - k_W v_{\xi}(\xi_S^-)]. \\ \text{I.C. : } \xi_S(t = 0) &= -1.\end{aligned}\tag{2.5}$$

Ice zone, $\xi_S \leq \xi < 1$,

$$\begin{aligned}\text{PDE : } v_t &= k_I \left[\frac{(1-\xi)^4}{4x_{\text{ref}}^2} v_{\xi\xi} - \frac{(1-\xi)^3}{2x_{\text{ref}}^2} v_{\xi} \right], \quad \xi_S < \xi < 1, \quad 0 < t. \\ \text{B.C. : } v(\xi = \xi_S, t) &= 0, \quad 0 \leq t, \\ v(\xi = 1^-, t) &= T_{\text{Ice}} < 0, \quad 0 \leq t.\end{aligned}\tag{2.6}$$

Formulation of the interface dynamics differs slightly from an early version (Appendix A in [12]). The formulation here is more suitable for extracting data on the finite interval.

We remark on both finite interval models that, (i) the spatial boundary $\xi = 1$, which corresponds to *infinity* in the semi-infinite interval, is a removable singularity;

Figure 2: Analytic solutions and temporal derivatives at eight equally-distanced times by exponential transform to finite interval at three different scales.

and (ii) the interface point is related to that in the semi-infinite interval by $\xi_s(t) = 1 - 2^{1-S(t)/x_{ref}}$ or $\xi_s(t) = (S(t) - x_{ref})/(S(t) + x_{ref})$ for *Model F1* and *Model F2*, respectively. The behavior of the solutions and their temporal derivatives are shown

in Figs.2 and 3 for the two models at several scales. We regard the scale factor as a control variable. It is important to computational accuracy and efficiency. Its effect in many respects are thoroughly examined in the sequel. We refer to [6, 7, 22] for more on transforms (mappings). Specialization to one-phase problem in a finite interval is omitted. It can be easily derived by setting the *far-field* temperature $v(\xi = 1^-) = 0$ as in the semi-infinite interval.

We note the continuous system, in the semi-infinite or finite interval, can be solved by numerical procedures in an algorithmically unified manner.

3. Spline Model vs Finite Difference Model at Nonuniform Nodes

Cubic spline interpolation and spline collocation (or FD method) are performed in the water and ice zones at every time steps in our computation. These two topics are well-documented in many places. We will briefly give the highlights, simply to help our detailed discussion in later sections.

We consider a node sequence, $\{x_0 < x_1 < \dots < x_{n-1} < x_n\}$, in each of water and ice zones, and set $h_i = x_i - x_{i-1}$, $1 \leq i \leq n$. Let $y = \{y_i \mid 0 \leq i \leq n\}$ be the interpolation data at the nodes and s be the (C^2 -smooth) cubic spline (piecewise polynomial) interpolant with knot sequence $\{x_i\}$. Conventional notations in FD (finite difference) method, such as $s_i \equiv s(x_i)$ and $s'_i \equiv s'(x_i)$, are adopted. Also, C-language notations for vectors and matrices are applied. Ocassionally, we abuse the notation and write y'' for s'' , also s for y . Nonuniform distributions of nodes are allowed. Specifically, we will investigate numerically the uniform and several orthogonal nodes. We refer to [4, 15] for general references.

3.1. Cubic Spline Interpolation at Nonuniform Nodes

By the construction of cubic spline interpolation, it can be derived that,

$$0 \leq i \leq n-1, \quad s'(x_i^+) = \frac{y_{i+1} - y_i}{h_{i+1}} - \frac{h_{i+1}}{3}s''_i - \frac{h_{i+1}}{6}s''_{i+1}, \quad (3.1)$$

$$1 \leq i \leq n, \quad s'(x_i^-) = \frac{y_i - y_{i-1}}{h_i} + \frac{h_i}{6}s''_{i-1} + \frac{h_i}{3}s''_i. \quad (3.2)$$

Two immediate consequences follow. By averaging the above, for $1 \leq i \leq n-1$,

$$s'(x_i) = \frac{1}{2} \left(\frac{y_{i+1} - y_i}{h_{i+1}} + \frac{y_i - y_{i-1}}{h_i} \right) - \frac{h_{i+1} - h_i}{6}s''_i + \frac{h_i}{12}s''_{i-1} - \frac{h_{i+1}}{12}s''_{i+1}, \quad (3.3)$$

and by subtraction, we obtain

Cubic spline interpolation algorithm (at interior nodes). For $1 \leq i \leq n-1$,

Figure 3: Analytic solutions and temporal derivatives at eight equally-distanced times by algebraic transform to finite interval at three different scales.

$$\begin{aligned}
 (Mw[i], Mp[i], Me[i]) (s''_{i-1}, s''_i, s''_{i+1})^T \\
 = (Fw[i], Fp[i], Fe[i]) (y_{i-1}, y_i, y_{i+1})^T, \quad (3.4)
 \end{aligned}$$

3.2. Cubic Spline Collocation at Nonuniform Nodes

Aiming at DM (differentiation matrix) approach to solving differential equations, we restrict our discussion to cubic spline model with *not-a-knot* boundary condition only. The first- and second-order DMs are denoted by D and $D2$, respectively.

3.2.1. Second-Order Differentiation Matrix

In the Stefan problem, Dirichlet boundary conditions are assumed at two ends in each of the water and ice zones. We obtain from equations (3.4, 3.6, 3.7) a linear algebraic system $Ms'' = Fy$, defined by tridiagonal matrices M and F of sizes $(n-1)$ -by- $(n-1)$ and $(n-1)$ -by- $(n+1)$, respectively. The averaging operator, M , is (strictly) diagonally dominant. The second-order differentiation matrix is therefore uniquely solvable that

$$s'' \equiv D2 \cdot s = M^{-1} \cdot F \cdot y,$$

which yields s''_i at interior nodes. We can determine s''_0 and s''_n by the B.C., equation (3.5). The differentiation matrix $D2[1 : n-1][1 : n-1]$ extends accordingly to $D2[0 : n][0 : n]$.

3.2.2. First-Order Differentiation Matrix

Once we obtain the second derivative of the spline, the first derivative is at hand. A pseudo-code reads like

$$\begin{aligned} D[0][\] &= -\frac{x_1-x_0}{3} D2[0][\] - \frac{x_1-x_0}{6} D2[1][\] + \frac{y_1-y_0}{h_1}, \\ D[n][\] &= \frac{x_n-x_{n-1}}{6} D2[n-1][\] + \frac{x_n-x_{n-1}}{3} D2[n][\] + \frac{y_n-y_{n-1}}{h_n}, \end{aligned}$$

and for $1 \leq i \leq n-1$,

$$\begin{aligned} D[i][\] &= \frac{h_i}{12} D2[i-1][\] - \frac{h_{i+1}-h_i}{6} D2[i][\] - \frac{h_{i+1}}{12} D2[i+1][\] \\ &\quad + \frac{1}{2} \left(\frac{y_{i+1}-y_i}{h_{i+1}} + \frac{y_i-y_{i-1}}{h_i} \right). \end{aligned}$$

Here the defaulted index, j , runs through $0 : n$. The above relations are implied by the C^2 -smoothness assumption (equations 3.1-3.3), independent of any B.C..

Remark 1. In practical implementation, explicit generation of the differentiation matrices can be avoided by applying a *reversed communication* technique, similar to the design of pre-conditioned iterative solvers in a sparse linear algebra package [20].

3.3. Spline Model vs Finite Difference Model

In FD modeling, the first and second derivatives are estimated at (interior) nonuniform nodes, respectively, as

$$y'_i = \frac{h_i}{h_i + h_{i+1}} \frac{y_{i+1} - y_i}{h_{i+1}} + \frac{h_{i+1}}{h_i + h_{i+1}} \frac{y_i - y_{i-1}}{h_i} - \frac{h_i h_{i+1}}{6} y_i^{(3)} - \frac{(h_{i+1} - h_i) h_i h_{i+1}}{24} y_i^{(4)} + HOTs \quad (3.8)$$

and

$$y''_i = \frac{\frac{y_{i+1} - y_i}{h_{i+1}} - \frac{y_i - y_{i-1}}{h_i}}{(h_{i+1} + h_i)/2} - \frac{h_{i+1} - h_i}{3} y_i^{(3)} - \frac{h_{i+1}^2 - h_{i+1} h_i + h_i^2}{12} y_i^{(4)} + HOTs. \quad (3.9)$$

These can be derived by a formal Taylor expansion argument. The implied three-point stencils are simplified at uniform nodes to standard central differences

$$y'_i = \frac{y_{i+1} - y_{i-1}}{2h}, \quad (0, 1, 0) (y''_{i-1}, y''_i, y''_{i+1})^T = \frac{1}{h^2} (1, -2, 1) (y_{i-1}, y_i, y_{i+1})^T.$$

These are to be compared with the cubic spline model. Equation (3.4) is a fourth-order accurate implicit relation, while the three-point stencil in equation (3.9) can be an implicit or explicit relation. Both schemes can reproduce the second derivative of a cubic polynomial at uniform nodes. However, the three-point difference in equation (3.9) may be inferior to second-order at nonuniform nodes, as we will see in Tables 20-22.

Summary. We conclude that cubic spline collocation is advantageous, in theory, mostly in case of nonuniform grid. It can be regarded as a compact FD scheme.

We mention in passing that, at uniform nodes, the compact scheme ([8])

$$\frac{1}{12} (1, 10, 1) \cdot (y''_{i-1}, y''_i, y''_{i+1})^T = \frac{1}{h^2} (-1, 2, -1) \cdot (y_{i-1}, y_i, y_{i+1})^T$$

is a fourth-order approximation to y'' . However, it needs four boundary values, y_0, y_n, y''_0 and y''_n . In our application to solving *two-point boundary value problems*, y_0 and y_n are given while y''_0 and y''_n are unknowns. Therefore, some kind of one-sided seven-point schemes are needed for y'' at the boundaries, and relatively very dense grid is needed for numerical stability. We ruled out the choice of this scheme in our numerical experiment.

4. Numerical Methods for the Stefan Problem

The finite interval *Stefan* problem, equations (2.1-2.3 or 2.4-2.6), consists of three subsystems

1. the water zone, with PDE and B.C. depending on a moving boundary.
2. the water-ice interface, with ODE and I.C. describing the dynamics of the interface.
3. the ice zone, with PDE and B.C. depending on a moving boundary.

Computational methods of solving a moving boundary problem fall into two categories : moving coordinate methods and fixed coordinate methods. We will not discuss the former, while the latter can be further calssified as being based on either dynamic or static grid. For clarity, we discuss the grid arrangements in these two approaches separately.

4.1. Discretization of the Geometry in Dynamic Grid Approach

Dynamic (time dependent) grid is adopted in the fixed coordinate system. At computational nodes, the relations

$$\begin{aligned}
 -1.0 = x_0 < x_1 < \dots < x_{n_W} < \xi_S(t), \\
 x_{n_W+1} &= \xi_S(t), \\
 \xi_S(t) < x_{n_W+2} < \dots < x_{n_W+1+n_I} < x_{n+1} \\
 &= 1.0,
 \end{aligned}$$

are maintained in the water zone, the interface and the ice zone, respectively. The numbers of interior nodes are n_W and n_I for the two zones. Through the whole simulation, the total number of interior nodes in the dynamic grid approach is kept a constant, $n = n_W + n_I + 1$. Two methods of geometry discretization are considered.

- FD. The grid is subuniform, i.e., uniform in both water and ice zones with probably different mesh sizes.
- DM. We apply either spectral or spline collocation in each of the water and ice zones, but not mixed. For spectral collocation, only Chebyshev extrema are deployed. Other choices are computationally inefficient in updating the collocation nodes and matrices at each time step. For spline collocation, we take both uniform and several orthogonal nodes. See Table 4 for all collocations compared and discussed later.

The FD or DM nodes and differentiation matrices in water and ice zones must be re-set at each time step. The number of nodes (and sizes of differentiation matrices) may need adjustment(s) as the interface moves.

4.2. Discretization of the Geometry in Static Grid Approach

Static (time independent) grid is adopted in the fixed coordinate system. The computational nodes are

$$\begin{aligned} -1.0 = x_0 < x_1 < \dots < x_{n_W} < \xi_S(t), \\ \xi_S(t) &\in (x_{n_W}, x_{n_W+1}), \\ \xi_S(t) < x_{n_W+1} < \dots < x_{n_W+n_I} < x_{n+1} = 1.0, \end{aligned}$$

in the water zone, the interface and the ice zone, respectively. The numbers of interior nodes are n_W and n_I for the two zones. The memory for the interface location, ξ_S , is allocated separately. The nodes are almost uniform in each zone, except near the interface ξ_S . Through the whole simulation in static grid approach, the total number of interior nodes is kept a constant, $n = n_W + n_I$. Similar to the case of dynamic grid, the interior nodes in each zone must be re-assigned at each time step. The numbers of interior nodes in each zone, n_W and n_I , may need to be adjusted as the interface moves in time. Our preliminary test shows that the DM method is not suitable in the static grid approach, probably because of the clustering of nodes near the two ends, with possible exception for spline collocation at uniform nodes. In static grid arrangement, only FD method is thoroughly examined.

4.3. Discretization of the Equations

In solving the ODE and two PDEs, two-level FD is applied to temporal derivatives, while spatial derivatives are resolved through either of the following.

1. FD.

We apply equations (3.8,3.9) to spatial derivatives in both water and ice zones, and adopt one-sided extrapolation in solving the interface dynamics (the Stefan condition). Tridiagonal version of LU works well.

2. Spectral or spline collocation via DM.

- (a) For spectral collocation, (shifted and scaled) Chebyshev extrema are taken as nodes in each zone. The nodes and DM are known in explicit analytic forms for this case. This enables efficient generations of the nodes

and DMs (and CMs, equations (4.1, 4.2)) at each time step. The choice of Chebyshev zeros can be converted to yield closed-end nodes, and in explicit expression as well. However, it takes a lot of computation to determine the DM with Chebyshev zeros. All other choices of spectral nodes demand a lot of computation in obtaining even the collocation nodes as solutions to algebraic eigenvalue problems, in addition to the generations of DMs.

- (b) For cubic spline collocation in each zone, the nodes can be uniform or orthogonal. Several choices of spline nodes are tested and discussed, as shown in Tables 4, 11-16, in which the hyperbolic-tangent function is used to generate distributions of nodes either clustered near both ends of a region (type *hyper2*) or clustered only near the interface (type *hyper1*). The Chebyshev and Legendre zeros are converted to closed-end nodes.
- (c) At each time step, the CM (collocation matrix) is

$$C = \frac{1}{\Delta t} I - k \left(\frac{\log 2}{x_{ref}} \right)^2 \left((1 - \xi)^2 D_2 - (1 - \xi) D \right) \quad (4.1)$$

by exponential transform (equations (2.1),(2.3)), and

$$C = \frac{1}{\Delta t} I - k \left(\frac{1}{x_{ref}} \right)^2 \left(\frac{(1 - \xi)^4}{4} D_2 - \frac{(1 - \xi)^3}{2} D \right) \quad (4.2)$$

by algebraic transform (equations (2.4),(2.6)), with ξ denoting interior nodes and $k = k_W$ (or k_I) in water (or ice) zone. The above subsystem in water or ice zone extends to boundary nodes by the corresponding boundary conditions.

Remark 2. The variable coefficients in the PDEs involve a term, $1 - \xi$, which is bounded above by 2 and decays *exponentially* (Subsection 2.1) or *algebraically* (Subsection 2.2) near the end node ($\xi = 1.0$) in ice zone. Therefore, equations (2.1) and (2.4) in water zone are more diffusion-dominant near $\xi = -1.0$, while equations (2.3) and (2.6) in ice zone more advection-dominant near $\xi = 1.0$.

4.4. More on Solving the PDEs

For spatial derivatives, either FD or DM method ([6],[7], [22]) is taken. Central differences (equations 3.8,3.9) are applied in the former case with first-order upwind difference of the advection term as an option. Spatial differentiations for subsystems in the water or ice zone can be implicit or explicit. In the implicit FD approach, the resulting

discrete algebraic subsystems can be solved by tridiagonal Gaussian elimination (GE) without pivoting or by iterative methods via three-point stencils. The two subsystems of PDEs are each treated as a two-point boundary value problem with Dirichlet data. An ODE equation of IVP type governs the evolution of position of the interface.

The discrete subsystems in spatial collocation approach, involving the Chebyshev or spline differentiation matrix, are solved by dense LU and BiCGSTAB [19] as a double check for some tests. The above-mentioned linear solvers are chosen based on thorough examinations.

Our preliminary test shows that implicit backward Euler scheme yields more accurate results than explicit forward Euler scheme. We note that MOL type approach for temporal discretization is a popular method for moving boundary problems, at least easy for the *one-phase* problem, but impractical for (static or dynamic grid) fixed coordinate approach to the *two-phase* problem in view of computational inefficiency due to the moving interface.

4.5. More on Solving the Interface Dynamics

One-sided spatial derivatives in the Stefan condition (equation (2.2) or (2.5)) are resolved at the interface by either of the following.

1. Spectral differentiation by DM.

Here the interface ODE is solved by two-level backward Euler scheme. The one-sided spatial derivatives are obtained using the first-order DM (Subsubsection 3.2.2) in each of the water and ice zones.

2. Extrapolation by FD.

Various one-sided schemes are possible for this purpose. We note the ODE at the interface is of first order in spatial variable, we expect in theory using a third order scheme so as to keep the over-all second order convergence in solving the Stefan problem. Our numerical experiment suggests usage of the four-point third-order extrapolation scheme of one-sided derivatives of S for better accuracy and stability. A pseudo-code for the water zone subsystem in static grid approach reads like

$$\begin{aligned}
 i &= n_W; \\
 du_1 &= (u_{i-0} - u_{i-1})/(x_{i-0} - x_{i-1}); & m_1 &= (x_{i-1} + x_{i-0})/2; \\
 du_2 &= (u_{i-1} - u_{i-2})/(x_{i-1} - x_{i-2}); & m_2 &= (x_{i-2} + x_{i-1})/2; \\
 du_3 &= (u_{i-2} - u_{i-3})/(x_{i-2} - x_{i-3}); & m_3 &= (x_{i-3} + x_{i-2})/2; \\
 u_x(S^-) &= +du_1 * (x_S - m_2) * (x_S - m_3)/(m_1 - m_2)/(m_1 - m_3) \\
 &\quad + du_2 * (x_S - m_1) * (x_S - m_3)/(m_2 - m_1)/(m_2 - m_3) \\
 &\quad + du_3 * (x_S - m_1) * (x_S - m_2)/(m_3 - m_1)/(m_3 - m_2);
 \end{aligned}$$

The notations are interpreted in an obvious manner. Similar argument holds for $u_x(S^+)$ and a simplified version works in dynamic grid approach. The same idea applies to both semi-infinite and finite intervals. We omit further algorithmic details in discretizing the *Stefan* condition. It is well-known now that low order finite element and finite volume methods are equivalent to finite difference method. Even a quadratic scheme in the finite volume approach to current problem leads to very complicated derivations ([29]). The Crank-Nicolson type variants were found not worthy of the double efforts in our preliminary test. The exponential type extrapolation suffers numerical instability and loses precision very soon.

4.6. Solution Procedure in Summary

In a time march, the whole discrete algebraic system consists of three blocks at each time step. The two PDEs with B.C. in water and ice zones are solved firstly. The interface dynamics (ODE with I.C.) is resolved secondly. Appropriate post-processing then ends the computation at a time step. Reversing the order in solving the PDEs and ODE yields practically identical results in a long-term simulation. In case of the FD method on either static or dynamic grid, robust and efficient tridiagonal system solvers are identified. Results are confirmed by using both direct and iterative methods. In case of dynamic grid and DM method, dense-version LU and BiCGSTAB method ([25]) outperform several other linear solvers. Numerical experiments of all the above is discussed in Section 5.

4.7. A Critical Common Issue in FD and DM Approaches: Interpolation

In both the dynamic and static grid approaches, interpolations are needed to update physical and geometric variables at the end of each time step. Among various options, the fourth-order accurate *not-a-knot* cubic spline interpolation is chosen for the current second-order PDE systems. Other choices might be using radial basis function or *Pade* approximation. However, the *local support* property ([4]) and therefore numerical stability of the spline model is preferred.

4.8. A Threshold Strategy for Static Grid

The application of a static grid is much more difficult and complicated than that of a dynamic grid. When the interface front approaches an integral computational node, the regular three-point FD approximation becomes close to being singular and is sensitive to truncation error. This inevitably leads to loss of significant digits. The propagation

of error can be fatal in a long-term simulation. To overcome this intrinsic difficulty, we designed a threshold strategy in which a frontal position, when approaching a computational node within a threshold, is re-assigned cross the integral node appropriately. Housekeepings are carried out accordingly, i.e., all relevant variables including the clock time are modified properly. This may yield, at the time step, a linearly compatible continuation of the physical process, provided that the threshold is roughly at the scale of the motion at one computational time step. In a sense, the threshold signifies the level of relative error. We note the idea is a specialization of *level set* methods ([14, 17, 21]). The threshold we used in all results shown later is $1.0e-06$ relative to the spatial mesh size. Some of our computational results are double-checked by using the value $1.0e-08$ instead. No significant difference is found by the *Unix* utility *diff* in detailed outputs.

4.9. Algorithm Complexity in DM Approach

In complexity analysis below, m denotes one of the three : $n_W, n_I, \max(n_W, n_I)$. We discuss only the scalar operation counts and refer to [3, 24, 27] for details at the code level.

- (I) Spline collocation. In addition to initial setup and overheads after the final time step, the simulation process consists of the following tasks.
 - (1) Solve the water zone problem. This is similar to the next step with much smaller subsystem in current application Table 3.
 - (2) Solve the ice zone problem using updated nodes and DMs.
 - (2.1) Set up the implicit system. The first two items on task list below may be saved if the simulation code is optimized and aimed at simple goals only.
 - (2.1a) Copy the first-order DM: an $\mathcal{O}(m^2)$ operation.
 - (2.1b) Copy the second-order DM: also $\mathcal{O}(m^2)$.
 - (2.1c) Process the diffusion term : an $\mathcal{O}(m)$ operation.
 - (2.1d) Process the advection term : an $\mathcal{O}(m)$ operation.
 - (2.1e) Process the temporal difference : an $\mathcal{O}(m)$ operation.
 - (2.1f) Assemble the CM : $\mathcal{O}(m)$, can be saved if carefully designed.
 - (2.2) Solve the system by dense LU. This is an $\mathcal{O}(m^3)$ operation. Precisely, $\mathcal{O}(m^3)$ for the forward elimination and $\mathcal{O}(m^2)$ for each backward substitution.

- (3) Solve the interface equation by two-level backward Euler scheme and two four-point third-order extrapolations. This calls for only as constant as many operations.
 - (4) Update as post-processing before end of each time step. Re-set geometry-dependent variables in water and ice zones, as follows.
 - (4.1) Set new nodes : $\mathcal{O}(m)$.
 - (4.2) Determine second-order DM by m tridiagonal solve. This is an $\mathcal{O}(m^2)$ task.
 - (4.3) Determine the first-order DM : $\mathcal{O}(m^2)$.
 - (4.4) Re-set physical variables by cubic spline interpolation in each zone. The tasks are to determine the spline (structure) and to evaluate new variables. Each of these tasks needs $\mathcal{O}(m)$ operations.
- (II) Spectral collocation. The solution procedures are very similar to the above. We note the following.
- (1) Operation counts in the generation of DMs. The complexity of the setup of first-order DM is $\mathcal{O}(m^2)$. But that of the second-order DM (as a matrix-matrix multiplication) is $\mathcal{O}(m^3)$, which is one order more expensive than the setup of cubic spline collocations.
 - (2) The condition numbers of CMs. These grow quickly on *Spectral* Chebyshev collocations, while the condition numbers remain moderate on spline collocations at uniform or orthogonal nodes.

Numerical observations are discussed later.

4.10. Algorithm Complexity in FD Approach

The operation counts of setup, solving and post-processing about the three-point stencils (and tridiagonal systems) are all linear ($\mathcal{O}(m)$) in the grid size in either the water or ice zone. Run-time efficiencies of various computational models are discussed later.

5. Numerical Experiments and Discussion

Our numerical experiments are motivated by an attempt to answer the following questions.

1. How to choose between using exponential and algebraic transforms ?

2. How to choose between static grid and dynamic grid approaches ?
3. How to choose between FD and DM methods ?
4. In application of DM method (on dynamic grid), how to choose between spectral and spline collocations ?
5. In application of FD method, how to choose between static grid and dynamic grid ?
6. In application of DM method on dynamic grid, how to choose the node distribution?
7. In application of FD method on dynamic grid, how to choose the node distribution?
8. In case of FD method on static (uniform) grid, what is the performance compared to other models ?
9. What are the scale effects in all the above ?
10. How do answers to the above change in long-term simulations ?
11. How do answers to the above change in much finer (or coarser) grids ?
12. Can we pinpoint a finite interval computational model whose performance is superior to or comparable with the best performance of models in the semi-infinite interval ? What is the specific combination of parameters and options ?

The question list is ordered in a *top-down* manner. However, the object-oriented tests have to be conducted in sort of a *bottom-up* sequence. Below we describe our experiments, relevant tables and figures, and some observations.

5.1. Computer Environment

Most numerical experiments reported here are running on a Tyan VX50B4985-E server system with AMD Opteron 8350 CPU at 2.0 Ghz. Other experiments are running on several Tyan servers of previous generations. The run time environment is practically single user and single task via remote *shell*. The software is designed in *C-language* and compiled by *gcc* with *O2* optimization. However, the codes serve for very general purposes so that the *cpu-time* of computations are not to be compared cross tables. Some technical subroutines (functions) are designed with prototypes from several references : [10, 18, 19, 24, 27]. The special function subroutines we used were tested and matched the known tabulated numbers [30].

5.2. Parameters and Computational Models

Physical parameters are shown in Table 1. These include the Stefan constant (α), the boundary temperatures, and the diffusivity constants (k_w , k_i) of water and ice zones respectively. These nondimensional constants are chosen for no particular reasons.

<i>Domain</i>	<i>Spatial – range</i>	k_w	k_i	α	T_{water}	T_{ice}	<i>time – span</i>
Finite	$[-1, 1]$	5.5967-02	2.0260-01	1.0	1.0	-1.0	[2,3],[2,12]
Semi-infinite	$[0, 10], [0, 20]$	5.5967-02	2.0260-01	1.0	1.0	-1.0	[2,3],[2,12]

Table 1: Physical parameters.

Domain truncations ([6, 22]) in test cases on semi-infinite models are based on Table 2, where it is shown that the choice of truncation at $x = 10.0$ or 20.0 enables using numerical B.C. at the far field accurate in 5 or 16 digits. The former, with approximately accurate boundary values, were used for general test purposes, while the latter was chosen in some *burning* test cases. The table also reflects the *similarity* effect. Accordingly, some tests were using $(t_0, t_{end}) = (2.0, 3.0)$ and $x_{max} = 10.0$, rather than $(2.0, 12.0)$ and 20.0 . The above are defaults in all tests, if not stated otherwise. Since all one-phase models are special cases of the corresponding two-phase models, with the far-field temperature equal to zero (the freezing temperature), we consider here only two-phase models. Actually, one-dimensional one-phase models are probably best tackled by moving coordinate approach.

The water and ice zone problem sizes of sample tests are recorded in Table 3 as reference for subsequent test runs.

Shown in Table 4 are two types of collocations adopted in the DM method, and also various nodes distributions for FD method. The abbreviations for nodes correspond to the *method-and-nodes* coupling (Table 5) in a one-to-one manner, and will be referred as such subsequently. We explain in some detail below.

1. Global approximation is based only on (first kind) Chebyshev extrema, i.e., two end-points plus zeros of a Chebyshev polynomial of the second kind. Spectral collocations at nodes other than Chebyshev extrema are tested and found too costly in cpu-time. Such results are not shown.
2. Local approximation using cubic spline model is confined at closed-end nodes such as uniform, Chebyshev extrema or zeros (converted). Other choices of

t_{end}	x_{max}	analytic-solution value	t_{end}	x_{max}
12.0	2.0	-5.5760888305236378e-01	3.0	1.0
12.0	4.0	-9.1540961863156045e-01	3.0	2.0
12.0	6.0	-9.9209835612639885e-01	3.0	3.0
12.0	8.0	-9.9965322317623884e-01	3.0	4.0
12.0	10.0	-9.9999300606173236e-01	3.0	5.0
12.0	12.0	-9.9999993602806814e-01	3.0	6.0
12.0	14.0	-9.999999973683862e-01	3.0	7.0
12.0	16.0	-9.999999999951583e-01	3.0	8.0
12.0	18.0	-9.9999999999956e-01	3.0	9.0
12.0	20.0	-1.0000000000000000e-00	3.0	10.0

Table 2: Dirichlet data towards infinity for domain truncation.

orthogonal nodes are denied because of stability or efficiency concerns after some preliminary tests. The results of (converted) Legendre zeros are presented in several tables only for illustrations.

We note that, the finite interval model with dynamic grid and *Spectral* collocation using algebraic transform is equivalent to a rational Chebyshev method ([6, 22]) applied to the original semi-infinite interval.

Computational models and numerical methods are summarized in Table 5. In the category of fixed coordinate approach, both FD and DM methods on dynamic grid are conducted. On static grid, only results in FD method (and uniform nodes) are presented. These are shown in the table with abbreviations for later reference. A test run is judged as *failed* if at any time step the interface position (S) turns backward. Computations of FD method on static grid at nonuniform nodes are unsatisfactory in our preliminary tests, probably due to the clustering of nodes at ends and instability of extrapolation. Similarly, DM method seems not appropriate on static grid with only possible exception for uniform nodes, in the author's experience.

$Domain$	t_{end}	n_x	$Transf$	x_{ref}	$initial$ $n_{x,W}$	$final$ $n_{x,W}$	$initial$ $n_{x,I}$	$final$ $n_{x,I}$
$[-1, 1]$	3.0	100	exp	0.5	25	29	75	71
$[-1, 1]$	3.0	100	exp	1.5	9	11	91	89
$[-1, 1]$	3.0	100	exp	2.5	6	7	94	93
$[-1, 1]$	3.0	100	alg	0.5	29	33	71	67
$[-1, 1]$	3.0	100	alg	1.5	12	15	88	85
$[-1, 1]$	3.0	100	alg	2.5	8	9	92	91
$[-1, 1]$	12.0	200	exp	0.5	49	99	151	101
$[-1, 1]$	12.0	200	exp	1.5	18	41	182	159
$[-1, 1]$	12.0	200	exp	2.5	11	26	189	174
$[-1, 1]$	12.0	200	alg	0.5	58	100	142	100
$[-1, 1]$	12.0	200	alg	1.5	24	50	176	150
$[-1, 1]$	12.0	200	alg	2.5	15	33	185	167
$[0, 20]$	12.0	2000			20	49	1980	1951

Table 3: Initial and final grid sizes in water vs ice zone. Subuniform nodes are assumed.

5.3. Experiments

Our discussion is focused on computation using the finite interval models. Results in the semi-infinite interval shown in this work serves for comparison purpose only.

As summarized in Table 6, we conducted several groups of numerical experiments:

- A. Validation of solvers of the three subsystems : Table 7.
- B. Reference cases using semi-infinite interval models : Tables 8- 10.
- C. Performance and condition numbers of various collocations : Tables 11, 12.
- D. Analysis of cpu-time profile for collocation models : Tables 13, 14.
- E. Effect of scale in collocation models : Tables 15, 16.
- F. Effect of scale and optimal scale in finite difference models : Tables 17- 19.
- G. Unit time simulations on dynamic grids : Tables 20- 22.
- H. Long-term simulations at various scales : Tables 23- 25.

<i>Type</i>	<i>Nodes</i>	<i>Abbreviation</i>
(local) spline collocation	uniform	spl-unif
(local) spline collocation	Legendre zeros	spl-Leg
(local) spline collocation	Chebyshev zeros	spl-Cheb1
(local) spline collocation	Chebyshev extrema	spl-Cheb2
(global) spectral collocation	Chebyshev extrema	Spectral
finite difference	uniform	unif
finite difference	Chebyshev zeros	Cheb1
finite difference	Chebyshev extrema	Cheb2
finite difference	Hyperbolic-tangent	hyper1
finite difference	Hyperbolic-tangent	hyper2

Table 4: Types of spatial discretizations and nodes.

- I. Finite difference models with small time steps in $[-1, 1]$ vs $[0, +\infty)$: Tables 26, 27.

Compared with the analytic solution (equation (2.12) in [12]), errors in the calculated solution at the final time and at all spatial nodes are reported in discrete *maximal* and *normalized 1-norm* in the e_{max} and e_{av} columns of tables. The *signed* error in the calculated interface position is shown in the e_s column. The accumulated cpu-time in the whole time marching, including overheads in pre- and post-processing within the time loop, appears in the *t-march* column.

5.4. Observations

We analyze the experiments and report some observations here. Firstly, in some other applications, dense LU with only partial pivoting may fail and GS iteration wanders. We examined carefully our subsystem linear solvers for all relevant computational models.

<i>Numerical Model</i>	<i>Domain</i>	<i>Transform</i>	<i>Grid Type</i>	<i>Spatial Discretization</i>	<i>Nodes in Water/Ice</i>
alg-stat-FD	Finite	algebraic	static	FD	(sub)uniform
exp-stat-FD	Finite	exponential	static	FD	(sub)uniform
alg-dyna-FD	Finite	algebraic	dynamic	FD	(non)uniform
exp-dyna-FD	Finite	exponential	dynamic	FD	(non)uniform
alg-dyna-DM	Finite	algebraic	dynamic	DM	Spectral or spl-*
exp-dyna-DM	Finite	exponential	dynamic	DM	Spectral or spl-*
stat-FD	Semi-infinite		static	FD	(sub)uniform
dyna-FD	Semi-infinite		dynamic	FD	(non)uniform
dyna-DM	Semi-infinite		dynamic	DM	Spectral or spl-*

Table 5: Computational models in fixed coordinate approach.

1. Various linear solvers were examined in DM approach on dynamic grid. We omit the table of results. The preferred methods are LU and BiCGSTAB in view of accuracy, numerical stability and run-time efficiency. The former was used in all subsequent computation, while the latter as a double check for some cases.
2. In case of FD approach, the PDEs in water and ice zones yield discrete tridiagonal linear systems. The GE method and GS in checkboard order are preferred for efficiency. We used the direct (GE) method in all further tests of FD models.
3. Table 7. In using the FD method, one-sided extrapolation scheme for the interface position (S) is critical in solving the interface dynamics. We tested several schemes with details shown here for two of them only : scheme 2 (three-point second-order) and scheme 3 (four-point third-order). The ratio $\Delta t/(\Delta x)^2$ are kept a constant (0.125) in all cases. Note that we adopt (general) central differences (equations 3.8,3.9) in the discretizations of PDEs in water and ice zones. We observed the following.
 - (a) Using the third-order extrapolation at uniform nodes, convergence of averaged system simulation error (e_{av}) is superior to second-order in spatial mesh size (cases Tu2-Tu4 vs Tu6-Tu8). While using the second-order extrapolation, the error is slightly inferior to second-order (cases S1-S4 vs S5-S8). Scheme 3 performs much better than scheme 2 on both static and dynamic grid using either transforms. Similar behaviors are seen with the other two measurements of errors (e_{max}, e_S). The effort (cpu-time) differs only a little in using either extrapolation scheme.

<i>Test goal</i>	<i>*-dyna-DM on $[-1,1]$</i>	<i>*-dyna-FD on $[-1,1]$</i>	<i>*-stat-FD on $[-1,1]$</i>	<i>*-FD on $[0,\infty)$</i>	<i>dyna-DM on $[0,\infty)$</i>
validation of linear solver	T*	T*			
extrapolation		T7	T7		
reference cases				T8,T9	T10
performance of collocations	T11,T12				
cpu-time profile analysis	T13,T14				
effect of scale	T15,T16	T17	T18,T19		
unit time span simulations	T20-T22	T20-T22			
long-term simulations	T24,T25	T23-T25	T23,T24		
FD with small time step		T26,T27	T26,T27	T26,T27	

Table 6: Test goals and tables. Computation on $[0, \infty)$ are on truncated domains. Denoted by T* are table of results not shown because of space.

- (b) For dynamic grid at nonuniform nodes (group TC,Th,TH). the convergence in e_{av} are mostly inferior but close to second-order using either transform, as predicted by equations (3.8,3.9).
 - (c) For long-term simulation ($t_{end} = 12.0$) using the third-order extrapolation at uniform nodes, the convergence in system error is still better than second-order using exponential transform on both static and dynamic grid (cases Lu5,6,7,8). It is about second-order using algebraic transform on dynamic grid (case Lu3 vs Lu4) and slightly inferior on static grid (case Lu1 vs Lu2).
 - (d) Summary. We use the four-point third-order extrapolation in all subsequent FD computation of the Stefan condition, i.e., equations (2.2,2.5). Results here seemingly favors the exponential transform to algebraic transform, at $x_{ref} = 1.0$ and using FD approach. However, it awaits thorough examinations in combinations of various parameters.
4. Tables 8-10. It is the purpose of writing this paper to investigate combination(s) of various options for better performance in computational models in the finite interval. Reference data in the semi-infinite interval models in long-term simulation are recorded in Table 8.
- (a) Cases 1-12 demonstrate about second-order convergence in average system error and justify partially our implementation of stat-FD and dyna-FD computational models in the semi-infinite interval.
 - (b) The system accuracies in domain truncation at $x_{max} = 10$ (cases 13-26) are inferior to accuracies in truncation at $x_{max} = 20$ (cases 1-12). This is certainly consistent with the observation made on Table 2.

- (c) Summary. We regard the marked cases (3,6,11) in the semi-infinite interval as *tough* reference cases for later comparisons. The fact that PDEs in the semi-infinite interval are (linear and) easier to solve, together with the limited precision one can achieve in computation of the *err* function makes these reference cases seemingly matchless. These are about optimal in balance between accuracy and efficiency.

As a first try, a few data obtained in finite interval models are shown in the last group in the table. The results of cases 28 and 30c are nearly comparable to the above *tough* reference cases in both accuracy and run-time efficiency. We will investigate more later.

Some more numerical evidence on correctness (of the convergence behavior) of code is given in Table 9. Collocation results in the semi-infinite interval are shown in Table 10, to demonstrate the seemingly matchlessness of the reference cases using FD in Table 8. The (small) condition numbers of the CMs are not to be blamed.

5. We start serious investigations of computational models in the finite interval by firstly looking at the computational performance of DM methods. Results are shown in Table 11. We note the following.
 - (a) By comparison of the average error, cases 1-4 vs 9-12 using exponential transform and cases 15-18 vs 27-30 using algebraic transform, it is found that all the *-dyna-DM models on spl-* nodes are about second-order in space while the Spectral collocations slightly better than second-order. Recall the not-a-knot cubic spline interpolation conducted at the end of each time step is *fourth-order* accurate and the PDEs are of second-order, the observed convergence is reasonable. This comment applies to other tables as well.
 - (b) Using exponential transform, the spline collocation at Chebyshev extrema is more accurate than at uniform nodes or Chebyshev zeros (case 3 vs 1,2 and 11 vs 9,10) in all three measurements of errors.
 - (c) Using algebraic transform, results at uniform nodes are more accurate than at Chebyshev extrema or zeros (case 15 vs 16,17 and 27 vs 28,29), also in all three measurements of errors.
 - (d) Spectral collocation *seemed* the most accurate and efficient at a first glance, should one compare cases 4 vs 5,6,7 and 18 vs 27,28,29 (or 19-21,23-25). The observation remains true for all three measurements of errors.

However, the calculation with Spectral models is less preferred with grid size larger than 100 or so. In those cases, the cpu-time increases eight times if the grid size doubles just in one-dimension. On the contrary, the increase in cpu-time is only five to six times in spline collocations.

- (e) Summary. Spectral collocation is much more accurate than spline collocations at as coarse as grid size 100. Beyond this, the cubic-order growth in the cost of cpu-time is a serious drawback for Spectral collocations. On the other hand, spline collocation models offer scalable performance. The two preferred subcases are exp-dyna-DM and alg-dyna-DM models, precisely, using exponential transform at Chebyshev extrema and using algebraic transform at uniform nodes.

The spl-Leg data (case 14) is included just to show its disadvantage : solving (two) algebraic eigenvalue problems at every time step is simply unaffordable in cpu-time.

6. Run-time efficiency and condition numbers of various collocation matrices are shown in Table 12. The condition numbers of the CMs in Spectral and spl-Cheb2 models grow with refined grids in a hand-in-hand manner. Thanks to the $\frac{1}{\Delta t}$ term in the diagonal of the CM, the quality of the computation remains acceptable. The cpu-time makes the difference. The Spectral model needs a dense linear solver (BiCGSTAB is not always reliable). The cpu-time spent for each of the three spline collocation modles is comparatively much smaller.
 - (a) Fact. The condition number reduces linearly as the temporal mesh refines in all type collocations, but grows in cubic-order of grid size in spectral collocation, while only second-order in spline collocations.
 - (b) The cpu-time increases linearly in n_t , while grows in cubic-order of n_x in spectral collocation, and only five to six times in spline collocations at doubled grid size.
 - (c) Summary. The three spline type collocations (at uniform nodes, Chebyshev zeros and extrema) are preferred to spectral collocation for efficiency, especially on a fine grid.

The results of cases in Legendre nodes showed very expensive cost in cpu-time.

7. For a further investigation, profile details of cpu-time spent in various DM methods are examined in unit time span and long-term simulations separately. The results are shown in Tables 13 and 14.

- (a) The simulation spent most time in solving the coupled system of equations. The pre and post out the *system solve* are minor, i.e., the ratio $(t_{nl-solve})/(t_{march})$ is close to 1.0, as can be derived from rightmost two columns of Table 13.
 - (b) The *system solve* consists of four steps : solve in the water zone, the ice zone, at the interface, and update *everything*. In a semi-implicit approach, solving the interface equation needs just two matrix-vector multiplications (via DM). The cost of such are not shown. Solving PDEs in the water and ice zones cost only (the same) little time for Spectral and spline collocations, as recorded in the $t - W$ and $t - I$ columns.
 - (c) The update cost in cpu-time are measured both within the calling subroutine (function) and also in the caller (main.c). It is seen that the compiler overheads in function calls are minor (the $t - update$ column vs the *update* column).
 - (d) The *update* process contains three steps : update (vectors and matrices) for water and ice zones, and a post-interpolation. In our application, the ice zone contains much more nodes than in the water zone. Setting up new DM in ice zone makes a very big difference between spectral and spline collocation matrices. As shown previously in an algorithmic complexity analysis (Subsection 4.9), the setup of a spectral DM is an $\mathcal{O}(m^3)$ operation, while only an $\mathcal{O}(m^2)$ operation for a spline DM. Here m denotes the number of nodes in water or ice zone. We can foresee that evenly distributed nodes in water and ice zones will not make much difference in the contrast of the comparison. We mention that the spline interpolation can be decomposed into two steps : (i) determine the representation structure and (ii) evaluate. We checked and found the vector version of the evaluation procedure saves only a tiny percentage of cpu-time.
 - (e) Summary. Spectral collocation, even the Chebyshev type, is not preferred in solving time-dependent PDEs at more than about 100 nodes, especially when the DMs need to be adjusted all the time.
8. Recorded in Table 15 are the effect of scale in spline collocations using both transforms at uniform nodes vs Chebyshev extrema.
- (a) Using exponential transform, Cheb2 nodes is superior to uniform nodes in both e_{av} and e_{max} at all scales but 0.2.
 - (b) Using algebraic transform, the error e_{av} (e_{max}) at Cheb2 nodes is smaller than at unif nodes for scales $x_{ref} > 2.5$ (1.5).

- (c) Actually, excluding the case $x_{ref} = 0.2$ (the only scale the water zone contains more nodes than ice zone at the initial time), the exp-dyna-DM model at Cheb2 nodes (the second group in table) outperforms all other three groups in the average error e_{av} , while the alg-dyna-DM model yields best calculation of the interface in most cases. The optimal scale in each respect do not necessarily require *even distribution* of nodes in the usual sense.
 - (d) Summary. As marked by *down arrows* ($\downarrow$), the errors e_{av} and e_{max} reduce with increasing scales until impractical situation (the water zone has too few nodes to start). However, the cpu-time in solving the ice zone and therefore the whole problem grows either. Tuning the control variable (the scale factor) remains a question. Visualization through Figs.(2, 3) does not explain everything.
9. Table 16 provides a comparison of spectral collocation with spline collocation in previous table, both at Chebyshev extrema. Spectral collocation is less stable. It yields more accurate result at the cost of much more cpu-time.
 10. Computational errors at various scales in FD method on dynamic grid at various nodes are studied with results shown in Table 17.
 - (a) The uniform nodes (cases 1-8,25-32) yield more accurate results than hyper1 nodes (cases 17-24,41-48). This is true in the average error (e_{av}) using either exponential or algebraic transform, for all tested scales. It is also true in the maximal error. As shown, the uniform nodes outperforms the hyper1 nodes in all respects but e_s .
 - (b) Excluding the extreme scale 0.2, the uniform nodes yield better results in e_{av} and e_{max} than Chebyshev extrema. On the contrary, the Cheb2 nodes provides result more accurate than uniform nodes in e_s using both transforms, with the exception $x_{ref} = 0.2, 2.0$. We note more nodes are clustered near x_s in the deployment of Cheb2 and hyper1 nodes. Therefore extrapolation yields better estimation of x_s .
 - (c) Summary. The Chebyshev extrema seems the best choice of nodes for *-dyna-FD models in minimizing the e_s error, while uniform nodes optimal in minimizing the average error, and also more accurate in e_{max} in most cases, except for scale factors 0.2 and 2.0. At both the uniform nodes and Chebyshev extrema, the average error decreases (as marked) with increasing scales.
 11. Recorded in Table 18 are scale effect in various aspects on static grid using FD models. We note the following.

- (a) The exp-stat-FD model is more accurate than alg-stat-FD model in e_{av} at all scales, except the extreme case $x_{ref} = 0.2$. It is interesting to note that with scale $x_{ref} = 0.2$, the nodes are about evenly distributed (100-to-99) in water and ice zones at initial time, while the average accuracies achieved in both models are the worst, compared to other scales. The scaling with $x_{ref} = 0.2$ is therefore regarded as improper.
 - (b) The diagonal dominance (rightmost four columns of table) increases with larger scales as expected from the PDEs (equations 2.1,2.3, 2.4,2.6). This seems not an important issue. The algebraic linear tridiagonal system can be solved in a stable way for all diagonally dominant matrices. Choosing economic temporal mesh size results in accurate computation.
 - (c) More regular behaviors can be seen in e_{max}, e_{av} , but less in e_S . This seems a characteristic of free boundary problems. To achieve minimal error (e_{max} , e_{av} or e_S), the optimal choice of x_{ref} are different in using algebraic or exponential transform. These are marked by asterisks (*) in the table.
 - (d) Summary. With ratio $\Delta t/(\Delta x)^2 = 20/8$ in this table, exponential transform is more suitable to our application than algebraic transform. The comment is consistent with findings discussed later in Tables 26, 27 and accompanies Table 25 giving a whole picture. The optimally scaled exponential transform (at $x_{ref} = 1.2$) yields more accurate results in e_{av} and e_{max} than algebraic transform at any scale. To one's surprise, it is not particularly useful to equally distribute the nodes in water and ice zones, neither the resulting diagonal dominance due to scaling.
12. Table 19. We take a closer look at the four *-*-FD models. The scale which minimizes any of the errors (e_{max}, e_{av}, e_S) in each of the four groups of test cases are marked by asterisks (*).
- (a) Summary. The results indicate that neither the maximal nor the average error is much influenced by the specific scale which optimized the interface error e_S .
 - (b) Still, we pose the important question : can we develop a feedback theory which predicts the over- or under-estimation of the interface position ? This may be important in two- or three-dimensional applications. We conjecture that residuals in the computations may shed light on this.
13. Reported in Tables 20- 22 are results, at three scales, of using finite interval models on dynamic grid in unit time span at three consecutively refined grid resolutions. At each scale, two groups of tests are conducted using different

transforms. The smallest error in each group are marked by an asterisk (*). Based on the accuracy and efficiency, we note the following.

- (a) All computational models on dynamic grid (*-dyna-FD and *-dyna-DM) exhibit about second-order convergence in the errors e_{av} and e_{max} . This is true for both transforms and all three scales.
 - (b) The winners are the *-dyna-FD models at uniform nodes. The corresponding average errors (e_{av}) are the least in each group. These are ranked as E1 and A1 in the last column of the three tables.
 - (c) We regard the *-dyna-FD models at Chebyshev extrema as the second, marked by E2 and A2. The performance of these models are, in general, next to the best.
 - (d) We reserved the third place for a spline collocation model. At scale 0.5, spline collocation at uniform nodes is selected for both transforms, while Chebyshev extrema is selected at scales 1.5 and 2.5.
 - (e) Summary. Computations by exp-dyna-FD model at uniform nodes yield the least average error, while exp-dyna-FD or exp-dyna-DM model at Chebyshev extrema yields the least error in the calculated interface position. Different applications may favor either choice. Spectral (Chebyshev) collocations at more than 100 nodes are inferior in the balance between accuracy and efficiency.
14. Table 23. Performances of four *-*-FD computational models in long-term simulation are investigated here. The temporal and spatial mesh sizes are chosen here to keep the cpu-time in *t-march* column comparable to our *tough* references in Table 8.
- (a) The first two groups of tests were running at scale 1.0. The first group cases show an saturation of accuracy in the alg-stat-FD model at $n_x = 400$ (cases 1-5), while the second group cases demonstrate significant progresses in exp-stat-FD model at the same temporal and spatial meshes (cases 11-15) as marked by *down arrows* ($\downarrow$). Here we focused on the average error as marked by $\downarrow$ in e_{av} column. It is noted that even case 11 ($n_t = 40000$) yields errors e_{av} and e_{max} smaller than case 5 ($n_t = 640000$). In view of this prominent performance, the resolution with $(n_t, n_x) = (320000, 400)$ from case 14 is therefore selected for further test.
 - (b) The third group of tests show that the performance of the exp-stat-FD model is stable, exhibiting similar behavior at various scales except for the (improper) extreme scale 0.5. A glance at Figs. 2(a) and 2(b) explains

why. We note many subcases in this group are only slightly less accurate than our *tough* reference cases in the semi-infinite interval (Table 8), while costing less cpu-time. The remaining cases (28-36) used *-dyna-FD models. The exp-dyna-FD model (cases 28-33) achieved slightly better accuracy than the exp-stat-FD model (the third group, cases 16-21) but at some more cpu-time.

- (c) Summary. In long-term simulation of a *natural* diffusive process in the semi-infinite interval, the transformed problem on the finite interval is better solved by using the exponential than using the algebraic transform, for a wide range of scales. The exp-stat-FD model is more accurate than alg-stat-FD model at the cost of about 10% more cpu-time, due to different sizes of ice zone. In long-term simulation, the exp-dyna-FD model may be a good substitute for exp-stat-FD model at uniform nodes. There exist uncertainties of risk in the implementation of threshold strategy in long-term simulations. With algebraic transform, the computational accuracy saturated at $n_x = 400$ and alg-dyn-FD model performs more accurate than alg-sta-FD model.

Finally, we note a comparison of cases 30,34,35,36 (two dynamic grid models at unif vs Cheb2 nodes). The accuracy at uniform nodes are better than at Chebyshev extrema, using either transform. This is the same as noted in unit time span simulations in Tables 20- 22.

15. Table 24. Long-term simulations ($t_{end} = 12.0$) of all models in the finite interval are conducted with $n_x = 100, 200, 500$. The ratio $\Delta t/(\Delta x)^2$ is fixed at $5/8$.
 - (a) For dynamic grid and FD method, using exponential transform (cases 2-6) is most accurate if deployed at uniform nodes (case 2). While using algebraic transform (cases 8-12), it is most advantageous at Chebyshev extrema (case 12). This is true in the averaged error and at two spatial resolutions ($n_x = 100, 200$). The best couplings of such (cases 2,12) is superior to FD on static grid using algebraic transform (case 7), and inferior to FD using exponential transform (case 1). This rule was observed here at scale 1.0.
 - (b) As far as the DM method is concerned.
 - i. With respect to e_{av} and e_{max} , the spectral collocation performs better using algebraic transform (case 20) than using exponential transform (case 16).
 - ii. On the contrary, the spline collocations yield smaller error using exponential transform (cases 13-15) than using algebraic transform (cases

17-19). Using either transform, spline collocation model is most accurate at uniform node distribution (cases 13,17).

iii. The spectral collocation (cases 16,20) is more accurate than all spline collocations, but the cpu-time grows in cubic-order in the spatial grid size. On the contrary, the growth is only five to six times with doubled grid size if using spline models.

(c) Summary. At the resolution $n_x = 200$, computations using exponential transform is more accurate than using algebraic transform. This holds for all three models (*-stat-FD,*-dyna-FD,*-dyna-DM) in their advantageous nodes distributions. These observations are still true for results (cases 21-24) on a finer grid with $n_x = 500$, whence the exponential transform favors uniform nodes and algebraic transform is doing good at Chebyshev extrema, as we noted in many instances.

16. Table 25. Computational models in the finite interval are compared in long-term simulations, at three scales : 0.5, 1.5 and 2.5. Very fine grids ($n_x = 1000$) are used in FD models (cases 1a-6c). For all cases in this table, the ratio $\Delta t/(\Delta x)^2$ ($=125/8$) indicates twenty-five times fewer time steps are taken in computation for cases here than cases in the previous and the next Tables 24, 26. With relatively large time steps here, the computational performance is better in using algebraic than using exponential transform. This holds for all grid sizes $n_x = 200, 400, 800, 1000$. Performance of cases are judged in the balance of (least) average error and (less) cpu-time.

(a) The first block (without case numberings) are all at scale 1.5. These showed computation using algebraic transform is superior to using exponential transform.

(b) The alg-stat-FD (alg-dyna-FD) model is more accurate than exp-stat-FD (exp-dyna-FD) model at the middle range scale (i.e., cases 4b vs 1b, 5b vs 2b). Actually, alg-*FD models are more accurate than the corresponding exp-*FD models in all but one case, i.e., cases 4a-4c better than 1a-1c, cases 5a-5b better than 2a-2b, cases 6a-6c better than 3a-3c.

(c) Only a few tests (cases 7b-15b) of DM models on coarse grid are shown. The increase of much more cpu-time on a denser grid is an unwanted disadvantage of collocations.

(d) Summary. Algebraic transform outperforms exponential transform in long-term simulation at the combination of very dense spatial grid and relatively large time steps. Actually, these are about optimal performance of models

in the finite interval and are slightly better than the *tough* references (Table 8) in the semi-infinite interval.

17. Table 26. Compared here are three computational models at three scales in eight groups of tests: the exp-stat-FD and alg-stat-FD models in the finite interval and the stat-FD model in the semi-infinite interval. The scale factors for finite interval models are 0.5, 1.5 and 2.5. All models are run in a unit time span at static subuniform nodes, except second half of the last group (case 8), which runs on dynamic subuniform nodes.
 - (a) The exp-stat-FD model is *the winner* in all respects. This model is more accurate than the other two in all spatial resolutions ($n_x/100 = 1, 2, 4, 8$) at each scale. Ranked as D1, case 1d is superior to cases 4d,7d,8d, and case 2d (ranked as D2) superior to cases 5d,7d,8d, ... , and similarly for cases nc,nb,na ($n=2,3$).
 - (b) At scale factor 2.5, the other model (alg-stat-FD) in the finite interval is more accurate than semi-infinite interval models on coarse grid (case 6b/6c vs cases 7b/7c and 8b/8c). However, the accuracy of computation in the finite interval by algebraic transform saturated at spatial grid that $n_x > 400$ and relatively fine time steps. This seems a characteristic (disadvantage) of the algebraic transform.
 - (c) It is worth noting that, in the semi-infinite interval, the static grid model is more accurate than the dynamic grid model, on coarse grid ($n_x = 200$) only (case 7b vs 8b). We do not try setting (different) optimal threshold for the continuation process at each run.
 - (d) All three models are more or less second-order accurate in space, while the finite interval model using exponential transform show probably more potential. However, we will not pursue the final time accuracy more than seven digits. After all, the long-term calculation involving the *err* function has its limitation ([18]). The residual reductions, not shown by limitation of space in the table, indicate partially that the calculations are on the right track. This is of value in case no analytic solution is known.
 - (e) Summary. In all cases here the ratio $\Delta t/(\Delta x)^2$ is $5/8$, while the ratio is $125/8$ in cases 1a-6c in Table 25. Time steps are twenty-five times denser in current table. With fine time meshes, using exponential transform is superior to using algebraic transform.
18. Table 27 is a long-term simulation ($t_{end} = 12.0$) version of Table 26. All parameters are the same on the finite interval.

- (a) Computational results using exponential transform are more accurate than using algebraic transform, at scales 1.5 and 2.5.
- (b) Computational models on the finite interval are flexible with many options in choosing various parameters. The computations on the (truncated) semi-infinite interval can not achieve the same accuracy as efficiently as in the finite interval.

However, these are not as economic as the optimal results from Table 25.

6. Conclusion

In this work, the classical two-phase Stefan problem in the semi-infinite interval $[0, \infty)$ is transformed to the finite interval $[-1, 1]$ by either an algebraic or exponential mapping, with a scaling factor as a control variable which is critical to computational accuracy and efficiency.

Three groups of computational models are investigated.

- (A) Finite difference methods in static grid arrangement using algebraic or exponential transform, denoted as alg-stat-FD and exp-stat-FD models.
- (B) Finite difference methods on dynamic grid. These are the alg-dyna-FD and exp-dyna-FD models.
- (C) Collocation methods on dynamic grid. including the alg-dyna-DM and exp-dyna-DM models.

For a complete description of a computational model, more factors such as the scale and the deployment of nodes are to be specified. We reserve the node types *Spectral* and *spl-Cheb2* for spectral and spline collocations, both using the Chebtshev extrema for computational nodes. Other orthogonal nodes are applied as well. In terms of computational performance, we compare the accuracies of various models in combinations of many factors, at (roughly) the same cpu-time spent in unit time span or long-term simulations.

To answer the questions raised in Section 5, we conclude that

1. In choosing between the two transforms at the start, the exponential one is found superior to the algebraic one in either unit time span (Tables 20- 22) or long-term simulation (Tables 23, 24), with relatively fine time steps ($\Delta t/(\Delta x)^2 = 5/8$, or $10/8$). On the contrary, the algebraic transform is the winner in balance of accuracy and efficiency, with larger time steps ($\Delta t/(\Delta x)^2 = 125/8$, Table 25).

(a) at static uniform nodes via alg transf.

(b) at static uniform nodes via exp transf.

(c) at dynamic uniform nodes via alg transf.

(d) at dynamic uniform nodes via exp transf.

(e) at dynamic Chebyshev extrema via alg transf.

(f) at dynamic Chebyshev extrema via exp transf.

Figure 4: Error on finite interval by FD at scale 0.5.

2. In comparison between static and dynamic grid arrangements, the static grid approach with the proposed threshold strategy and implicit algebraic solve provides with the most accurate results among all models in finite interval. Although, the dynamic grid approach (with implicit algebraic solve) is also accurate and stable.

3. Various collocations are uniquely classified by node types (*Spectral* or *spl*-*). Among these, the *Spectral* collocation is the most accurate and effective (only) at grid size about 100. The corresponding cpu-time grows in cubic-order in the grid size, while it grows five to six times in spline collocations. The second-order convergence in spline collocations is stable and scalable.
4. At the bottom line, the FD method is more economic in achieving good accuracy than the DM method. At the *right* scale, the maximal and average errors are the least in the *-stat-FD models (finite difference on static grid). The errors are comparatively the worst in the *-dyna-DM models (collocations on dynamic grid). Ranked in between are the *-dyna-FD models (finite difference on dynamic grid). These are true in using either exponential or algebraic transform. Furthermore, the accuracy in using the formal transform is better than using the latter. On the other hand, the convergence in time of the maximal and average errors are the smoothest on dynamic grid by collocations and least smooth with possible zig-zagging on static grid by finite differences, due to the presence of the moving interface. The computational models on dynamic grid by finite differences stand in between again.
5. A closer look at Table 3 reveals the fact that the two-phase Stefan problem, in the semi-infinite interval, starts initially too close to the left end and therefore the ice region contains too large portion of the whole grid. While in the finite interval we can have better balance in the grid arrangement by choosing the control (scale) factor properly. In this sense, finite interval models can be better than semi-infinite ones. The problem size in the ice region can be smaller in finite interval. This means more economic computation and also smaller propagated error. The condition number of DM in the ice region can be much smaller than in the semi-infinite interval. Computations using the finite interval models are satisfactory and provide many options for various considerations. Such are preferred to those of semi-infinite interval models.
6. Different convergence behaviors are visualized in Figs.4, 5. It highlights several points.
 - (a) For current application, the scale 1.5 is a *right* scale in that the maximal and average errors decay in time as shown in Figs.5(b), 5(d), 5(f) and 5(c). This reflects the diffusive feature of the PDEs. These sub-figures also show that the convergence of errors in using the exponential transform is significantly smaller than using the algebraic transform.
 - (b) The spatial grid sizes are $n_x = 200$ in all tests shown in these figures. It

(a) at static uniform nodes via alg transf.

(b) at static uniform nodes via exp transf.

(c) at dynamic uniform nodes via alg transf.

(d) at dynamic uniform nodes via exp transf.

(e) at dynamic Chebyshev extrema via alg transf.

(f) at dynamic Chebyshev extrema via exp transf.

Figure 5: Error on finite interval by FD at scale 1.5.

is obviously seen for the *improper* scale, 0.5, that more spatial nodes (and more cpu-time) are needed.

Some of our observations on the numerical experiments can be explained to some extent by the graphical behaviors of the analytic solution and its deriva-

tive. To gain more knowledge for real-life problems, more theoretic and computational studies are needed.

7. Optimal scales (x_{ref}) with respect to the three measurements of errors (e_{av} , e_{max} , e_s) are different and depend on node types. Refined spatial and temporal mesh-sizes are not critical in minimizing the error e_s . Instead, a proper choice of the scale can make a big difference. This is true for both spatial FD and DM methods. It would be nice to know whether the calculated x_s is over- or underestimated. Such a feedback can be used to control for many purposes.

At the time of writing this paper, we have made successful simulations for a *real-life* problem using a mushy layer model [2, 13], (only) by the fixed coordinate static grid approach using the proposed threshold strategy in computational model in the semi-infinite interval. The findings here will hopefully shed light on more efficient, accurate and robust computation in the finite interval and in two- and three-dimensional applications.

Acknowledgments

The author thanks Tunghai university for laboratory support in a long period of time.

References

- [1] D.V. Alexandrov, Solidification with a quasiequilibrium two-phase zone, *Acta Mater*, **49** (2001), 759-764.
- [2] D.V. Alexandrov, D.L. Aseev, I.G. Nizovtseva, H.-N. Huang, D. Lee, Nonlinear dynamics of directional solidification with a mushy layer. Analytic solutions of the problem, *Int. J. Heat and Mass Transfer*, **50** (2007), 3616-3623.
- [3] R. Baltensperger and M.R. Trummer, Spectral differencing with a twist, *SIAM J.Sci.Comput.*, **24** (2002), 1465-1487.
- [4] Carl de Boor, *A Practical Guide to Splines, Revised Edition*, Springer-Verlog, NewYork (2001).
- [5] Carl de Boor, *Convergence of cubic spline interpolation with the not-a-knot condition*, MRC **2876**, University of Wisconsin at Madison (1985).
- [6] J.P. Boyd, *Chebyshev and Fourier-Spectral Methods, 2nd ed.*, Dover, Mineola, New York (2001).
- [7] C. Canuto, M.Y. Hussaini, A. Quarteroni, T.A. Zang, *Spectral Methods, Fundamentals in Single Domains*, Springer New York (2006).
- [8] L. Collatz, *The Numerical Treatment of Differential Equations*, Springer-Verlag, New York (1966).
- [9] J. Crank, *Free and Moving Boundary Problems*, Clarendon Press, Oxford, England (1984).
- [10] A. Greenbaum, *Iterative methods for solving linear systems*, SIAM, Philadelphia, USA (1997).
- [11] E. Javierre, C. Vuik, F.J. Vermolen, and S. van der Zwaag, A comparison of numerical models for one-dimensional Stefan problem, *JCAM*, **192** (2006), 445-459.
- [12] D. Lee, D.V. Alexandrov, Numerical Modeling of One-dimensional Binary Solidation - The Classical Two-phase Stefan Problem, *IJPAM*, **58**, No.4 (2010), 381-416.
- [13] D. Lee , D.V. Alexandrov and H.N. Huang, Numerical Modeling of Binary Solidification with A Mushy Layer, *International Journal of NM-TMA.*, **5**, No.2 (2012), 157-185.

- [14] Zhilin Li and Kazufumi Ito, *The Immersed Interface Method, Numerical Solutions of PDEs Involving Interfaces and Irregular Domains*, SIAM, Philadelphia, USA (2006).
- [15] G. Micula and S. Micula, *Handbook of Splines*, kluwer Academic Publishers (1999).
- [16] J. Morison, M. McPhee, R. Muench, et al., The LeadEx Experiment, *EOS. Trans.*, AGU **74** (1993), 393-397.
- [17] S. Osher, R. Fedkiw, *Level Set Methods and Dynamic Implicit Surfaces*, Springer, New York (2003).
- [18] W.H. Press, S.A. Teukolsky, W.T. Vetterling, B.P. Flannery, *Numerical recipes in C*, Cambridge University Press (1992).
- [19] Y. Saad, *Iterative methods for sparse linear system. 2nd ed.*, SIAM, Philadelphia, USA (2003).
- [20] Mark Seager, *A SLAP for the Masses*, Technical Report: UCRL-100267, Lawrence Livermore National Laboratory (1988).
- [21] J.A. Sethian, *Level Set Methods and Fast Marching Methods*, Cambridge University press, New York (1999).
- [22] Jie Shen, Tao Tang and Li-Lian Wang, *Spectral Methods - Algorithms, Analysis and Applications*, Springer-Verlag Berlin Heidelberg (2011).
- [23] I. Stefan, Uber Einige Probleme der Theorie der Warmeleitung, *Sitzber. Kais. Akad. Wiss. Wien, Math.-Naturw. Kl.*, **98**, No. 11a (1898), 437438.
- [24] L.N. Trefethen, *Spectral methods in MATLAB*, SIAM, Philadelphia, PA (2000).
- [25] H.A. van der Vorst, BiCGSTAB: A fast and smoothly converging variant of Bi-CG for the solution of nonsymmetric linear systems, *SIAM J. Sci. Comput.*, **13** (1992), 631-644.
- [26] C. Vuik, Some historical notes about the Stefan problem, *Nieuw Archief voor Wiskunde, 4e serie*, **11**, 157 (1993).
- [27] J.A.C. Weideman and S.C. Reddy, A MATLAB differentiation matrix suite, *ACMTOMs*, **26** (2000), 465-519.
- [28] J.S. Wettlaufer, M.G. Worster, H.E. Huppert, Solidification of Leads: Theory, Experiment and Field Observations, *J. Geophys. Res.*, **105** (2000), 1123-1134.

- [29] M. Zerroukat and C.R. Chatwin, *Computational Moving Boundary Problems*, Research Studies Press LTD., Taunton, Somerset England (1994).
- [30] Shanjie Zhang, Jianming Jin, *Computation of Special Functions*, John Wiley & Sons, Inc., New York, NY. USA (1996).

A. More Tables

Table 7: Effect of extrapolation scheme in interface evolution of *-*.FD models. $x_{ref} = 1.0, \Delta t/(\Delta x)^2 = \frac{1}{2}$.

<i>Case</i>	<i>Sch</i>	<i>t_{end}</i>	<i>n_t</i>	<i>n_x</i>	<i>Tr</i>	<i>Grid Type</i>	<i>Node</i>	<i>e_{max}</i>	<i>e_{av}</i>	<i>e_S</i>	<i>t- march</i>
S1	2	3.0	20000	100	alg	stat	unif	2.866-4	6.645-5	1.062-4	2.3
S2	2	3.0	20000	100	exp	stat	unif	3.145-4	4.068-5	1.044-4	2.6
S3	2	3.0	20000	100	alg	dyna	unif	6.260-4	9.276-5	2.300-4	2.5
S4	2	3.0	20000	100	exp	dyna	unif	5.678-4	7.216-5	1.910-4	2.9
S5	2	3.0	80000	200	alg	stat	unif	6.710-5	1.913-5	2.369-5	17.7
S6	2	3.0	80000	200	exp	stat	unif	8.739-5	1.108-5	2.762-5	20.7
S7	2	3.0	80000	200	alg	dyna	unif	1.734-4	2.488-5	6.155-5	19.8
S8	2	3.0	80000	200	exp	dyna	unif	1.585-4	1.949-5	5.131-5	22.8
Tu1	3	3.0	20000	100	alg	stat	unif	9.959-5	3.430-5	-8.665-6	2.3
Tu2	3	3.0	20000	100	exp	stat	unif	1.201-5	4.624-6	4.749-6	2.6
Tu3	3	3.0	20000	100	alg	dyna	unif	1.038-4	2.376-5	4.174-5	2.5
Tu4	3	3.0	20000	100	exp	dyna	unif	7.134-5	1.264-5	2.762-5	2.9
Tu5	3	3.0	80000	200	alg	stat	unif	3.811-5	1.130-5	-8.059-6	17.7
Tu6	3	3.0	80000	200	exp	stat	unif	3.007-6	1.011-6	4.309-7	20.7
Tu7	3	3.0	80000	200	alg	dyna	unif	2.010-5	5.001-6	8.214-6	19.8
Tu8	3	3.0	80000	200	exp	dyna	unif	1.542-5	2.669-6	6.041-6	23.1
TC1	3	3.0	20000	100	alg	dyna	Cheb2	9.874-5	1.548-5	-2.942-7	2.6
TC2	3	3.0	80000	200	alg	dyna	Cheb2	2.479-5	4.197-6	-1.227-6	20.7
TC3	3	3.0	20000	100	exp	dyna	Cheb2	5.353-5	1.132-5	-8.406-6	3.1
TC4	3	3.0	80000	200	exp	dyna	Cheb2	2.110-5	3.652-6	-4.684-6	23.8
Th1	3	3.0	20000	100	alg	dyna	hyper1	1.009-4	1.862-5	4.583-6	2.7
Th2	3	3.0	80000	200	alg	dyna	hyper1	2.551-5	4.605-6	1.042-6	21.2
Th3	3	3.0	20000	100	exp	dyna	hyper1	1.265-4	2.841-5	-2.643-5	3.0
Th4	3	3.0	80000	200	exp	dyna	hyper1	3.312-5	7.307-6	-7.062-6	23.6
TH1	3	3.0	20000	100	alg	dyna	hyper2	1.672-4	2.268-5	-8.650-7	2.7
TH2	3	3.0	80000	200	alg	dyna	hyper2	4.195-5	5.729-6	-5.720-7	21.0
TH3	3	3.0	20000	100	exp	dyna	hyper2	1.230-4	2.157-5	-2.665-5	3.1
TH4	3	3.0	80000	200	exp	dyna	hyper2	3.339-5	5.631-6	-7.568-6	24.2
Lu1	3	12.0	200000	100	alg	stat	unif	1.028-4	3.684-5	1.330-5	21.5
Lu2	3	12.0	800000	200	alg	stat	unif	2.756-5	1.012-5	1.053-6	172.7
Lu3	3	12.0	200000	100	alg	dyna	unif	1.107-4	3.393-5	3.730-5	24.6
Lu4	3	12.0	800000	200	alg	dyna	unif	2.776-5	8.479-6	9.669-6	195.4
Lu5	3	12.0	200000	100	exp	stat	unif	8.447-5	8.175-6	5.017-6	25.2
Lu6	3	12.0	800000	200	exp	stat	unif	2.044-5	1.671-6	4.941-7	197.9
Lu7	3	12.0	200000	100	exp	dyna	unif	8.555-5	1.804-5	2.743-5	27.8
Lu8	3	12.0	800000	200	exp	dyna	unif	2.055-5	3.897-6	5.798-6	218.3

Table 8: Reference cases in semi-infinite interval by FD. $t_{end} = 12.0$. Finite interval models are using FD and uniform nodes, $x_{ref} = 1.0$, $ratio = \Delta t / (\Delta x)^2$. Cases 3,6 and 11 are regarded as *tough* references.

<i>Case</i>	$x-range$	<i>Tr</i>	x_{ref}	n_t	n_x	<i>Ratio</i>	<i>Grid Type</i>	e_{max}	e_{av}	e_S	$t-march$
1	[0,20]			6250	500	1.0	stat	1.713-4	9.378-6	-5.638-5	3.2
2	[0,20]			25000	1000	1.0	stat	3.536-5	2.076-6	-1.024-5	26.0
3*	[0,20]			100000	2000	1.0	stat	7.815-6	4.836-7*	-2.025-6	*207.6
4	[0,20]			6250	500	1.0	dyna	8.565-5	7.327-6	1.827-5	3.7
5	[0,20]			25000	1000	1.0	dyna	1.129-5	1.345-6	1.027-5	29.9
6*	[0,20]			100000	2000	1.0	dyna	1.712-6	2.665-7*	3.924-6	*241.1
7	[0,20]			31250	500	5.0	stat	1.446-4	6.870-6	-6.846-5	16.3
8	[0,20]			125000	1000	5.0	stat	2.871-5	1.451-6	-1.327-5	131.1
9	[0,20]			500000	2000	5.0	stat	6.129-6	3.268-7	-2.781-6	1043.6
10	[0,20]			31250	500	5.0	dyna	2.070-5	3.092-6	6.691-6	18.8
11*	[0,20]			125000	1000	5.0	dyna	7.434-6	5.295-7*	7.264-6	*150.0
12	[0,20]			500000	2000	5.0	dyna	4.337-6	1.405-7	3.157-6	1197.1
13	[0,10]			6250	250	1.0	stat	1.713-4	1.901-5	-5.638-5	1.8
14	[0,10]			25000	500	1.0	stat	3.536-5	4.426-6	-1.024-5	14.1
15	[0,10]			100000	1000	1.0	stat	7.815-6	1.262-6	-2.025-6	113.4
16	[0,10]			200000	1000	0.5	stat	6.845-6	1.069-6	-2.498-6	222.6
17	[0,10]			6250	250	1.0	dyna	8.571-5	1.474-5	1.823-5	2.0
18	[0,10]			25000	500	1.0	dyna	1.130-5	2.927-6	1.027-5	16.2
19	[0,10]			100000	1000	1.0	dyna	6.845-6	8.189-7	3.924-6	129.2
20	[0,10]			200000	1000	0.5	dyna	6.845-6	6.202-7	3.445-6	257.3
21	[0,10]			31250	250	0.2	stat	1.446-4	1.401-5	-6.846-5	8.9
22	[0,10]			125000	500	0.2	stat	2.871-5	3.186-6	-1.327-5	71.1
23	[0,10]			500000	1000	0.2	stat	6.845-6	9.531-7	-2.781-6	567.2
24	[0,10]			31250	250	0.2	dyna	2.059-5	6.408-6	6.655-6	9.9
25	[0,10]			125000	500	0.2	dyna	7.428-6	1.335-6	7.261-6	80.4
26	[0,10]			500000	1000	0.2	dyna	6.845-6	5.794-7	3.157-6	643.9
27	[-1,1]	exp	1.0	320000	400	10.0/8	stat	5.042-6	5.634-7	3.053-7	157.5
28*	[-1,1]	exp	1.0	320000	500	15.6/8	stat	3.199-6	4.600-7*	2.941-7	*195.8
29	[-1,1]	exp	1.0	320000	800	40.0/8	stat	1.198-6	3.402-7	2.954-7	312.6
30a	[-1,1]	exp	1.0	160000	1000	125.0/8	stat	1.271-6	5.689-7	6.066-7	190.3
30b	[-1,1]	exp	1.5	102400	800	62.5/8	stat	2.216-6	8.607-7	6.199-7	103.5
30c*	[-1,1]	exp	1.5	204800	800	62.5/8	stat	1.221-6	4.974-7*	2.669-7	*205.3
30d	[-1,1]	exp	1.5	204800	800	62.5/8	dyna	1.926-6	6.555-7	7.271-7	230.6
30e	[-1,1]	alg	1.5	102400	800	62.5/8	stat	2.919-6	7.959-7	3.575-7	88.9
30f	[-1,1]	alg	1.5	204800	800	62.5/8	stat	2.530-6	7.446-7	-2.256-8	178.8
30g	[-1,1]	alg	1.5	204800	800	62.5/8	dyna	1.563-6	6.565-7	9.048-7	202.6

Table 9: Performance of FD method in the semi-infinite interval.

t_{end}	x_{max}	n_t	n_x	$Grid$ $Type$	e_{max}	e_{av}	e_S	$t-$ $march$
3.0	5.0	4000	100	stat	4.443-04	3.087-05	-1.192-04	1.6
3.0	5.0	16000	200	stat	9.195-05	7.012-06	-2.132-05	8.9
3.0	5.0	64000	400	stat	1.841-05	1.780-06	-3.972-06	67.2
3.0	5.0	4000	100	dyna	6.644-04	4.474-05	-1.631-04	1.7
3.0	5.0	16000	200	dyna	5.725-05	5.925-06	-1.172-05	10.0
3.0	5.0	64000	400	dyna	6.627-06	1.070-06	2.169-06	76.3
3.0	10.0	16000	200	stat	4.338-04	1.451-05	-1.238-04	8.3
3.0	10.0	64000	400	stat	8.909-05	3.178-06	-2.245-05	61.1
3.0	10.0	16000	200	dyna	6.517-04	2.048-05	-1.674-04	9.3
3.0	10.0	64000	400	dyna	5.163-05	2.404-06	-1.283-05	70.7
12.0	10.0	160000	200	stat	2.266-04	2.088-05	-1.213-04	81.6
12.0	10.0	640000	400	stat	4.544-05	4.595-06	-2.334-05	647.7
12.0	10.0	160000	200	dyna	8.168-05	1.115-05	-4.133-05	109.6
12.0	10.0	640000	400	dyna	1.088-05	1.680-06	6.461-06	742.2

Table 10: Performance of DM method on dynamic grid in the semi-infinite interval with $n_t = 16000$, $n_x = 200$.

$DM-$ $type$	e_{max}	e_{av}	e_S	$t-$ $march$	$cond_W$ $initial$	$cond_W$ $final$	$cond_I$ $initial$	$cond_I$ $final$
spl-unif	6.096-05	4.777-06	-1.400-05	146.4	1.02	1.01	1.1	1.1
spl-Cheb1	1.861-05	4.359-06	2.888-06	147.8	1.01	1.01	20.8	21.0
spl-Cheb2	5.049-05	5.632-06	1.250-05	149.8	1.02	1.01	58.3	58.9
Spectral	1.275-05	8.555-07	1.542-06	407.2	1.02	1.01	63.5	64.1

Table 11: Run-time efficiency and accuracy of collocation (*-dyna-DM) models at various nodes. $t_{end} = 3.0$, $x_{ref} = 1.0$, $ratio = \Delta t / (\Delta x)^2$.

<i>Case</i>	n_t	n_x	<i>Ratio</i>	<i>Tr</i>	<i>DM- Node</i>	e_{max}	e_{av}	e_s	t_{march}
1	20000	100	1/8	exp	spl-unif	5.657-5	1.217-5	-1.313-5	23.5
2	20000	100	1/8	exp	spl-Cheb1	4.680-5	1.275-5	-9.583-6	23.8
3	20000	100	1/8	exp*	spl-Cheb2*	3.682-5	1.144-5	-6.469-6	23.7
4*	20000	100	1/8	exp	Spectral	1.614-5	2.724-6	-1.769-7	45.6
5	20000	200	4/8	exp	spl-unif	2.476-5	4.721-6	-2.951-6	131.1
6	20000	200	4/8	exp	spl-Cheb1	1.792-5	4.385-6	-6.934-7	133.3
7	20000	200	4/8	exp	spl-Cheb2	1.710-5	4.289-6	-4.351-7	132.9
8	20000	200	4/8	exp	Spectral	1.146-5	2.101-6	1.284-6	356.1
9	80000	200	1/8	exp	spl-unif	1.666-5	3.238-6	-4.001-6	530.4
10	80000	200	1/8	exp	spl-Cheb1	9.662-6	2.934-6	-1.754-6	529.3
11	80000	200	1/8	exp*	spl-Cheb2*	8.838-6	2.840-6	-1.500-6	530.9
12	80000	200	1/8	exp	Spectral	3.202-6	5.696-7	2.180-7	1412.5
13	10000	100	2/8	alg	Spectral	2.385-5	4.601-6	2.482-6	19.9
14	20000	100	1/8	alg	spl-Leg	8.087-5	3.103-5	-2.137-5	346.4
15	20000	100	1/8	alg*	spl-unif*	5.833-5	2.139-5	-1.503-5	20.8
16	20000	100	1/8	alg	spl-Cheb1	8.166-5	3.101-5	-2.098-5	21.0
17	20000	100	1/8	alg	spl-Cheb2	8.337-5	3.111-5	-2.049-5	21.1
18*	20000	100	1/8	alg	Spectral	1.296-5	2.467-6	8.901-7	39.7
19	20000	200	4/8	alg	spl-unif	2.555-5	7.257-6	-3.497-6	116.9
20	20000	200	4/8	alg	spl-Cheb1	2.723-5	9.349-6	-3.899-6	115.3
21	20000	200	4/8	alg	spl-Cheb2	2.727-5	9.395-6	-3.905-6	116.8
22	20000	200	4/8	alg	Spectral	1.110-5	2.215-6	1.539-6	322.1
23	40000	200	2/8	alg	spl-unif	2.020-5	6.206-6	-4.285-6	229.8
24	40000	200	2/8	alg	spl-Cheb1	2.177-5	8.309-6	-4.695-6	233.8
25	40000	200	2/8	alg	spl-Cheb2	2.181-5	8.356-6	-4.703-6	232.9
26	40000	200	2/8	alg	Spectral	5.632-6	1.119-6	7.412-7	645.3
27	80000	200	1/8	alg*	spl-unif*	1.752-5	5.683-6	-4.678-6	459.6
28	80000	200	1/8	alg	spl-Cheb1	2.048-5	7.790-6	-5.093-6	466.5
29	80000	200	1/8	alg	spl-Cheb2	2.071-5	7.837-6	-5.101-6	465.9
30	80000	200	1/8	alg	Spectral	2.901-6	5.725-7	3.421-7	1295.0

Table 12: Condition numbers of CMs in *-dyna-DM models at various nodes.
 $t_{end} = 3.0$, $x_{ref} = 1.0$.

n_t	n_x	Tr	$DM-Node$	$t-march$	$Cond_W_{init}$	$Cond_W_{final}$	$Cond_I_{init}$	$Cond_I_{final}$
1000	100	exp	spl-unif	1.5	3.84e+0	3.98e+0	9.00e+0	8.39e+0
1000	100	exp	spl-Cheb1	1.6	6.89e+0	9.95e+0	8.63e+2	7.39e+2
1000	100	exp	spl-Cheb2	1.5	1.35e+1	2.13e+1	2.35e+3	2.01e+3
1000	100	exp	spl-Leg	19.9	5.80e+0	8.09e+0	6.21e+2	5.32e+2
1000	100	exp	Spectral	12.0	1.45e+1	2.30e+1	2.55e+3	2.19e+3
<hr/>								
2000	100	exp	spl-unif	3.1	2.40e+0	2.47e+0	4.86e+0	4.58e+0
2000	100	exp	spl-Cheb1	3.1	3.80e+0	5.24e+0	3.99e+2	3.41e+2
2000	100	exp	spl-Cheb2	3.1	6.92e+0	1.06e+1	1.09e+3	9.37e+2
2000	100	exp	spl-Leg	39.8	3.29e+0	4.37e+0	2.86e+2	2.45e+2
2000	100	exp	Spectral	23.9	7.41e+0	1.14e+1	1.19e+3	1.02e+3
1000	200	exp	spl-unif	8.1	1.41e+1	1.47e+1	4.02e+1	3.72e+1
1000	200	exp	spl-Cheb1	8.2	1.03e+2	1.64e+2	1.88e+4	1.61e+4
1000	200	exp	spl-Cheb2	8.2	2.53e+2	4.15e+2	5.14e+4	4.39e+4
1000	200	exp	spl-Leg	141.6	7.80e+1	1.23e+2	1.35e+4	1.16e+4
1000	200	exp	Spectral	93.9	2.74e+2	4.50e+2	5.57e+4	4.76e+4
<hr/>								
2000	200	exp	spl-unif	16.3	7.28e+0	7.56e+0	1.92e+1	1.78e+1
2000	200	exp	spl-Cheb1	16.2	4.82e+1	7.65e+1	8.65e+3	7.39e+3
2000	200	exp	spl-Cheb2	16.4	1.19e+2	1.94e+2	2.37e+4	2.03e+4
2000	200	exp	spl-Leg	280.3	3.67e+1	5.72e+1	6.19e+3	5.30e+3
2000	200	exp	Spectral	188.0	1.29e+2	2.11e+2	2.57e+4	2.20e+4
1000	100	alg	spl-unif	1.4	6.70e+0	6.66e+0	1.04e+1	8.98e+0
1000	100	alg	spl-Cheb1	1.4	2.29e+1	3.08e+1	1.05e+3	8.25e+2
1000	100	alg	spl-Cheb2	1.4	5.16e+1	7.22e+1	2.86e+3	2.24e+3
1000	100	alg	spl-Leg	17.6	1.82e+1	2.39e+1	7.59e+2	5.94e+2
1000	100	alg	Spectral	10.5	5.57e+1	7.82e+1	3.11e+3	2.43e+3
<hr/>								
2000	100	alg	spl-unif	2.8	3.77e+0	3.75e+0	5.52e+0	4.84e+0
2000	100	alg	spl-Cheb1	2.8	1.13e+1	1.49e+1	4.86e+2	3.81e+2
2000	100	alg	spl-Cheb2	2.8	2.48e+1	3.44e+1	1.33e+3	1.04e+3
2000	100	alg	spl-Leg	35.2	9.06e+0	1.17e+1	3.49e+2	2.74e+2
2000	100	alg	Spectral	21.0	2.68e+1	3.73e+1	1.44e+3	1.13e+3
1000	200	alg	spl-unif	7.3	2.95e+1	2.92e+1	5.03e+1	4.25e+1
1000	200	alg	spl-Cheb1	7.3	4.15e+2	5.79e+2	2.30e+4	1.80e+4
1000	200	alg	spl-Cheb2	7.2	1.05e+3	1.49e+3	6.26e+4	4.90e+4
1000	200	alg	spl-Leg	124.8	3.09e+2	4.28e+2	1.65e+4	1.29e+4
1000	200	alg	Spectral	82.4	1.14e+3	1.62e+3	6.79e+4	5.31e+4
<hr/>								
2000	200	alg	spl-unif	14.4	1.43e+1	1.42e+1	2.37e+1	2.02e+1
2000	200	alg	spl-Cheb1	14.4	1.92e+2	2.68e+2	1.06e+4	8.26e+3
2000	200	alg	spl-Cheb2	14.5	4.90e+2	6.95e+2	2.89e+4	2.26e+4
2000	200	alg	spl-Leg	247.1	1.43e+2	1.98e+2	7.58e+3	5.92e+3
2000	200	alg	Spectral	164.7	5.31e+2	7.54e+2	3.14e+4	2.45e+4

Table 13: cpu-time profile of various collocation (*-dyna-DM) models in unit time span. $t_{end} = 3.0, n_x = 100, n_t = 10000$.

x_{ref}	Tr	$DM-$ $type$	$update$ $-W$	$update$ $-I$	$update$ $-intrap$	$update$	$t-$ $update$	$t-$ W	$t-$ I	$t-nl$ $-solve$	$t-$ $march$
1.0	exp	spl-unif	0.31	6.76	0.22	7.40	7.44	0.12	5.60	13.37	16.83
1.0	exp	spl-Cheb2	0.31	6.73	0.22	7.45	7.49	0.12	5.57	13.40	16.79
1.0	exp	Spectral	0.22	*24.37	0.23	25.02	*25.06	0.12	5.67	31.08	34.49
1.5	exp	spl-unif	0.19	7.59	0.23	8.12	8.16	0.08	7.00	15.47	19.18
1.5	exp	spl-Cheb2	0.19	7.51	0.23	8.11	8.15	0.08	6.55	15.00	18.59
1.5	exp	Spectral	0.12	*28.25	0.23	28.82	*28.86	0.08	6.50	35.69	39.31
1.0	alg	spl-unif	0.46	6.28	0.22	7.07	7.11	0.18	5.28	12.80	15.92
1.0	alg	spl-Cheb2	0.45	6.16	0.22	7.02	7.06	0.18	4.92	12.37	15.37
1.0	alg	Spectral	0.43	21.84	0.23	22.71	22.75	0.19	5.51	28.71	31.82
2.0	alg	spl-unif	0.20	7.58	0.23	8.11	8.15	0.08	7.06	15.52	18.95
2.0	alg	spl-Cheb2	0.20	7.60	0.23	8.22	8.26	0.08	7.14	15.71	19.08
2.0	alg	Spectral	0.13	*28.07	0.23	28.63	*28.67	0.08	6.46	35.45	38.80

Table 14: cpu-time profile of various collocation (*-dyna-DM) models in long-term simulation. $t_{end} = 12.0, (n_x, n_t) = (100, 16000), (200, 64000), x_{ref} = 1.0$.

n_x	Tr	DM- type	$t-W$	$t-I$	$t-update$	$t-nl$ -solve	$t-march$	n_W initial	n_W final	n_I initial	n_I final
100	exp	spl-unif	1.67	28.96	43.45	75.02	97.84	13	29	87	71
100	exp	spl-Cheb1	1.67	28.97	46.20	77.90	100.51				
100	exp	spl-Cheb2	1.67	28.67	46.11	77.49	100.24	same	as	above	
100	exp	Spectral	1.66	28.51	*98.15	129.34	151.84				
100	alg	spl-unif	2.47	24.93	41.10	69.54	81.28	17	33	83	67
100	alg	spl-Cheb1	2.38	24.77	44.11	72.27	83.63				
100	alg	spl-Cheb2	2.44	25.22	44.50	73.23	85.15	same	as	above	
100	alg	Spectral	2.36	24.79	*87.19	115.35	126.92				
200	exp	spl-unif	33.65	856.04	696.16	1611.32	1905.66	26	58	174	142
200	exp	spl-Cheb1	33.51	854.79	746.07	1661.06	1957.17				
200	exp	spl-Cheb2	34.63	866.04	750.56	1678.37	1978.13	same	as	above	
200	exp	Spectral	32.36	822.47	*2931.68	3810.63	4095.15				
200	alg	spl-unif	50.46	725.40	649.59	1447.89	1572.52	34	66	166	134
200	alg	spl-Cheb1	50.78	734.56	699.52	1513.24	1637.00				
200	alg	spl-Cheb2	52.23	742.16	707.96	1556.92	1682.43	same	as	above	
200	alg	Spectral	47.51	701.64	*2544.85	3316.05	3432.00				

Table 15: Effect of scale in computational errors, subproblem sizes and condition numbers of spline collocations with $(n_t, n_x) = (4000, 200)$.

x_{ref}	Tr	N_{nodes} $spl-*$	e_{max}	e_{av}	e_S	$t-$ $march$	n_W $init$	n_W $final$	n_I $final$	$Cond_W$ $init$	$Cond_W$ $final$	$Cond_I$ $final$
0.2	exp	unif	1.44-2	4.01-4↓	3.57-4	32.0	101	115	85	1.1+2	1.0+2	5.8+1
0.5	exp	unif	5.89-5	2.01-5↓	9.52-6	42.0	49	58	142	1.4+1	1.4+1	2.6+1
0.8	exp	unif	6.44-5	1.49-5↓	4.59-6	51.6	32	39	161	5.9+0	6.0+0	1.3+1
1.0	exp	unif	6.79-5	1.30-5↓	2.65-6	57.1	26	32	168	4.1+0	4.2+0	9.0+0
1.5	exp	unif	7.68-5	9.91-6↓	-2.51-7	62.9	18	22	178	2.4+0	2.4+0	4.9+0
2.0	exp	unif	9.00-5	8.24-6↓	-2.67-6	69.1	14	17	183	1.8+0	1.8+0	3.3+0
2.5	exp	unif	1.18-4	7.91-6↓	-6.30-6	69.8	11	14	186	1.5+0	1.5+0	2.5+0
3.0	exp	unif	1.49-4	6.49-6↓	-7.48-6	75.2	10	12	188	1.4+0	1.4+0	2.1+0
3.5	exp	unif	1.51-4	7.29-6	-8.21-6	76.7	8	10	190	1.2+0	1.3+0	1.8+0
4.0	exp	unif	1.97-4	8.13-6	-1.17-5	75.6	7	9	191	1.2+0	1.2+0	1.6+0
0.2	exp	Cheb2	5.97-3	4.19-4↓	2.89-4↓	34.5	101	115	85	3.9+4	5.2+4	1.6+4
0.5	exp	Cheb2	5.87-5	1.94-5↓	1.02-5↓	43.8	49	58	142	1.0+3	1.5+3	2.1+4
0.8	exp	Cheb2	6.13-5	1.45-5↓	6.23-6↓	55.7	32	39	161	1.4+2	2.3+2	1.3+4
1.0	exp	Cheb2	6.12-5	1.22-5↓	5.25-6↓	58.6	26	32	168	5.6+1	9.1+1	9.4+3
1.5	exp	Cheb2	6.04-5	8.78-6↓	3.85-6↓	64.2	18	22	178	1.2+1	1.8+1	4.9+3
2.0	exp	Cheb2	6.01-5	7.02-6↓	3.00-6↓	69.1	14	17	183	4.6+0	6.4+0	2.9+3
2.5	exp	Cheb2	6.24-5	6.16-6↓	2.09-6↓	70.8	11	14	186	2.3+0	3.3+0	1.9+3
3.0	exp	Cheb2	6.46-5	5.46-6↓	1.42-6↓	77.3	10	12	188	1.9+0	2.2+0	1.4+3
3.5	exp	Cheb2	6.95-5	5.57-6	7.55-7↓	78.0	8	10	190	1.4+0	1.6+0	1.0+3
4.0	exp	Cheb2	8.34-5	6.00-6	-6.11-7↓	76.0	7	9	191	1.2+0	1.4+0	7.7+2
0.2	alg	unif	9.83-5	1.91-5	2.15-5	16.8	101	111	89	2.3+2	2.3+2	2.2+1
0.5	alg	unif	6.01-5	2.05-5↓	7.99-6	20.7	58	66	134	2.9+1	2.8+1	1.9+1
0.8	alg	unif	6.50-5	1.74-5↓	4.56-6	26.6	41	47	153	1.1+1	1.1+1	1.3+1
1.0	alg	unif	6.84-5	1.59-5↓	2.80-6	28.6	34	40	160	7.4+0	7.3+0	1.0+1
1.5	alg	unif	7.63-5	1.30-5↓	-5.87-8	36.0	24	29	171	3.7+0	3.8+0	6.2+0
2.0	alg	unif	7.69-5	1.06-5↓	-2.44-7	38.2	19	22	178	2.6+0	2.5+0	4.3+0
2.5	alg	unif	8.58-5	9.52-6↓	-1.88-6	39.0	15	18	182	2.0+0	1.9+0	3.3+0
3.0	alg	unif	1.07-4	9.03-6↓	-5.00-6	40.3	13	16	184	1.7+0	1.7+0	2.7+0
3.5	alg	unif	1.26-4	8.85-6↓	-7.21-6	41.0	11	14	186	1.5+0	1.5+0	2.3+0
4.0	alg	unif	1.16-4	7.75-6↓	-5.15-6	43.6	10	12	188	1.4+0	1.4+0	2.0+0
0.2	alg	Cheb2	1.87-4	2.25-5↓	2.60-5↓	16.7	101	111	89	8.8+4	1.1+5	7.6+3
0.5	alg	Cheb2	6.71-5	2.42-5↓	5.28-6↓	20.8	58	66	134	3.4+3	4.4+3	1.5+4
0.8	alg	Cheb2	7.04-5	2.00-5↓	3.04-6↓	26.7	41	47	153	5.7+2	7.3+2	1.3+4
1.0	alg	Cheb2	7.10-5	1.78-5↓	2.48-6↓	28.7	34	40	160	2.3+2	3.2+2	1.1+4
1.5	alg	Cheb2	7.15-5	1.39-5↓	1.74-6↓	36.1	24	29	171	4.5+1	6.9+1	6.6+3
2.0	alg	Cheb2	7.16-5	1.14-5↓	1.37-6↓	38.2	19	22	178	1.6+1	2.0+1	4.5+3
2.5	alg	Cheb2	7.23-5	9.86-6↓	1.01-6↓	38.8	15	18	182	6.3+0	8.7+0	3.2+3
3.0	alg	Cheb2	7.44-5	8.94-6↓	5.48-7↓	40.1	13	16	184	3.8+0	5.5+0	2.3+3
3.5	alg	Cheb2	7.75-5	8.30-6↓	5.56-8↓	41.3	11	14	186	2.4+0	3.5+0	1.8+3
4.0	alg	Cheb2	7.71-5	7.61-6↓	9.24-8	43.5	10	12	188	1.9+0	2.3+0	1.4+3

Table 16: Effect of scale in Spectral collocations.

x_{ref}	Tr	$\frac{n_t}{1000}$	$\frac{n_x}{100}$	e_{max}	e_{av}	e_s	t_{march}	n_W <i>init</i>	n_W <i>final</i>	n_I <i>final</i>	$Cond_W$ <i>init</i>	$Cond_W$ <i>final</i>	$Cond_I$ <i>final</i>
0.5	exp	2	1	1.10-4	2.78-5	2.41-5	7.4	25	29	71	1.3+2	1.6+2	2.2+3
1.0	exp	2	1	1.15-4	2.07-5	1.26-5	10.1	13	16	84	7.4+0	1.1+1	1.0+3
1.5	exp	2	1	1.26-4	1.79-5	6.21-6	11.2	9	11	89	2.3+0	3.0+0	5.4+2
2.0	exp	2	1	1.63-4	1.82-5	-1.82-6	12.1	7	9	91	1.5+0	1.9+0	3.2+2
2.2	exp	2	1	1.45-4	1.60-5	1.11-6	12.4	7	8	92	1.5+0	1.5+0	2.7+2
2.5	exp	2	1			failed		6					
0.5	exp	8	1	2.77-5	7.03-6	5.88-6	27.7	25	29	71	2.9+1	3.7+1	4.9+2
1.0	exp	8	1	3.26-5	5.73-6	1.95-6	38.4	13	16	84	2.5+0	3.4+0	2.2+2
1.5	exp	8	1	4.42-5	5.86-6	-1.22-6	44.2	9	11	89	1.3+0	1.5+0	1.2+2
2.0	exp	8	1			failed		7					
2.2	exp	8	1			failed		7					
2.5	exp	8	1			failed		6					
0.5	exp	8	2	2.74-5	7.04-6	6.07-6	182.8	49	58	142	5.2+2	7.7+2	1.0+4
1.0	exp	8	2	2.80-5	5.16-6	3.42-6	273.5	26	32	168	2.9+1	4.7+1	4.7+3
1.5	exp	8	2	2.94-5	4.30-6	2.09-6	306.1	18	22	178	6.5+0	9.6+0	2.5+3
2.0	exp	8	2	3.25-5	3.92-6	1.08-6	327.5	14	17	183	2.9+0	3.8+0	1.5+3
2.5	exp	8	2	4.04-5	3.99-6	-2.21-7	348.8	11	14	186	1.7+0	2.2+0	9.7+2
3.0	exp	8	2	4.82-5	3.72-6	-1.19-6	393.0	10	12	188	1.5+0	1.6+0	6.8+2
1.0	alg	1	1	2.20-4	4.33-5	3.11-5	2.0	17	20	80	5.6+1	7.8+1	2.4+3
1.0	alg	1	2	2.19-4	4.39-5	3.19-5	16.3	34	40	160	1.1+3	1.6+3	5.3+4
1.0	alg	4	2	5.48-5	1.10-5	7.93-6	64.7	34	40	160	2.5+2	3.5+2	1.1+4
1.0	alg	4	4	5.47-5	1.10-5	7.99-6	702.5	67	79	321	4.9+3	7.1+3	2.5+5

Table 17: Effect of scale in computational errors of *-dyna-FD models at various nodes. $t_{end} = 3.0, n_t = 4000, n_x = 200, \Delta t/(\Delta x)^2 = 20/8$.

<i>Case</i>	<i>Grid Type</i>	<i>Tr</i>	<i>FD- Nodes</i>	x_{ref}	e_{max}	e_{av}	e_S
1	dyna	exp	unif	0.2	1.285e-02↓	4.503e-04↓	-1.302e-04
2	dyna	exp	unif	0.5	4.748e-05↓	1.358e-05↓	1.548e-05
3	dyna	exp	unif	0.8	4.057e-05↓	1.070e-05↓	1.348e-05
4	dyna	exp	unif	1.0	3.534e-05↓	9.173e-06↓	1.285e-05
5	dyna	exp	unif	1.2	3.082e-05↓	7.983e-06↓	1.218e-05
6	dyna	exp	unif	1.5	2.316e-05↓	6.662e-06↓	1.169e-05
7	dyna	exp	unif	1.8	1.624e-05↓	5.691e-06↓	1.409e-05
8	dyna	exp	unif	2.0	1.437e-05↓	5.299e-06↓	1.163e-05
9	dyna	exp	Cheb2	0.2	5.912e-03	2.552e-04↓	9.685e-05
10	dyna	exp	Cheb2	0.5	6.175e-05	1.792e-05↓	8.577e-06
11	dyna	exp	Cheb2	0.8	6.664e-05	1.445e-05↓	4.303e-06
12	dyna	exp	Cheb2	1.0	7.193e-05	1.330e-05↓	1.981e-06
13	dyna	exp	Cheb2	1.2	7.805e-05	1.264e-05↓	1.083e-07
14	dyna	exp	Cheb2	1.5	8.777e-05	1.209e-05↓	-1.855e-06
15	dyna	exp	Cheb2	1.8	9.356e-05	1.149e-05↓	-1.704e-06
16	dyna	exp	Cheb2	2.0	1.007e-04	1.143e-05↓	-3.058e-06
17	dyna	exp	hyper1	0.2	2.319e-02	1.094e-03	-7.367e-04
18	dyna	exp	hyper1	0.5	1.004e-04	2.062e-05	1.044e-05
19	dyna	exp	hyper1	0.8	6.818e-05	1.901e-05	3.721e-06
20	dyna	exp	hyper1	1.0	7.931e-05	1.980e-05	-2.720e-07
21	dyna	exp	hyper1	1.2	9.285e-05	2.101e-05	-3.695e-06
22	dyna	exp	hyper1	1.5	1.192e-04	2.329e-05	-8.072e-06
23	dyna	exp	hyper1	1.8	1.520e-04	2.605e-05	-1.172e-05
24	dyna	exp	hyper1	2.0	1.753e-04	2.767e-05	-1.413e-05
25	dyna	alg	unif	0.2	8.785e-05↓	1.497e-05↓	1.697e-05
26	dyna	alg	unif	0.5	4.020e-05↓	1.191e-05↓	1.746e-05
27	dyna	alg	unif	0.8	3.563e-05↓	1.030e-05↓	1.593e-05
28	dyna	alg	unif	1.0	3.027e-05↓	9.102e-06↓	1.586e-05
29	dyna	alg	unif	1.2	2.704e-05↓	8.133e-06↓	1.507e-05
30	dyna	alg	unif	1.5	1.584e-05↓	6.772e-06↓	1.615e-05
31	dyna	alg	unif	1.8	1.804e-05	6.199e-06↓	1.679e-05
32	dyna	alg	unif	2.0	1.636e-05	5.480e-06↓	1.418e-05
33	dyna	alg	Cheb2	0.2	1.589e-04	1.489e-05↓	2.130e-05
34	dyna	alg	Cheb2	0.5	5.365e-05	1.428e-05↓	1.390e-05
35	dyna	alg	Cheb2	0.8	5.500e-05	1.356e-05↓	8.929e-06
36	dyna	alg	Cheb2	1.0	5.956e-05	1.308e-05↓	6.333e-06
37	dyna	alg	Cheb2	1.2	6.480e-05	1.264e-05↓	4.074e-06
38	dyna	alg	Cheb2	1.5	7.214e-05	1.230e-05↓	1.619e-06
39	dyna	alg	Cheb2	1.8	8.140e-05	1.219e-05↓	-2.859e-07
40	dyna	alg	Cheb2	2.0	8.885e-05	1.200e-05↓	-2.002e-06
41	dyna	alg	hyper1	0.2	3.211e-04	2.278e-05	2.163e-05
42	dyna	alg	hyper1	0.5	7.194e-05	1.270e-05	1.991e-05
43	dyna	alg	hyper1	0.8	4.489e-05	1.325e-05	1.273e-05
44	dyna	alg	hyper1	1.0	5.252e-05	1.444e-05	8.657e-06
45	dyna	alg	hyper1	1.2	6.124e-05	1.551e-05	5.109e-06
46	dyna	alg	hyper1	1.5	7.580e-05	1.745e-05	6.643e-07
47	dyna	alg	hyper1	1.8	9.295e-05	1.942e-05	-3.097e-06
48	dyna	alg	hyper1	2.0	1.065e-04	2.042e-05	-5.450e-06

Table 18: Effect of scale in computational errors, subproblem size and diagonal dominance (abbreviated as dd in last four columns) of *-stat-FD models.

$t_{end} = 3.0, (n_t, n_x) = (4000, 200), \Delta t/(\Delta x)^2 = 20/8.$

<i>Grid Type</i>	<i>Tr</i>	<i>x_{ref}</i>	<i>e_{max}</i>	<i>e_{av}</i>	<i>e_S</i>	<i>n_W init</i>	<i>n_W final</i>	<i>n_I init</i>	<i>n_I final</i>	<i>dd_W init</i>	<i>dd_W final</i>	<i>dd_I init</i>	<i>dd_I final</i>
stat	exp	0.2	1.28-2	4.35-4	-1.27-4	100	114	99	85	0.08	0.07	0.08	0.10
stat	exp	0.5	2.04-5	4.95-6	1.29-5	48	57	151	142	0.33	0.32	0.19	0.20
stat	exp	0.8	1.49-5*	4.77-6	8.94-6	31	38	168	161	0.55	0.55	0.32	0.34
stat	exp	1.0	1.58-5	4.66-6	7.21-6	25	31	174	168	0.66	0.66	0.41	0.42
stat	exp	1.2	1.72-5	4.61-6*	5.88-6	21	26	178	173	0.74	0.73	0.49	0.50
stat	exp	1.5	2.02-5	4.65-6	4.29-6	17	21	182	178	0.82	0.81	0.59	0.60
stat	exp	1.8	2.34-5	4.81-6	2.93-6	14	18	185	181	0.86	0.86	0.66	0.67
stat	exp	2.0	2.79-5	4.94-6	2.14-6	13	16	186	183	0.88	0.88	0.71	0.71
stat	exp	2.2	3.06-5	5.06-6	1.39-6	12	14	187	185	0.90	0.90	0.74	0.75
stat	exp	2.5	3.91-5	5.44-6	*2.49-7	10	13	189	186	0.92	0.92	0.78	0.79
stat	exp	2.8	4.83-5	5.76-6	-7.52-7	9	11	190	188	0.94	0.94	0.82	0.82
stat	exp	3.0	5.37-5	6.09-6	-1.58-6	9	11	190	188	0.95	0.95	0.84	0.84
stat	exp	3.2	6.56-5	6.41-6	-2.17-6	8	10	191	189	0.95	0.95	0.85	0.86
stat	exp	3.5	7.96-5	6.85-6	-3.10-6	7	9	192	190	0.96	0.96	0.87	0.88
stat	exp	3.8	9.19-5	7.22-6	-4.15-6	7	8	192	191	0.97	0.97	0.89	0.89
stat	exp	4.0	1.07-4	7.74-6	-4.93-6	6	8	193	191	0.97	0.97	0.90	0.90
stat	alg	0.2	8.04-5	2.50-5	*-1.82-8	100	110	99	89	0.04	0.04	0.15	0.21
stat	alg	0.5	2.35-5	1.11-5	7.82-6	57	65	142	134	0.19	0.19	0.20	0.24
stat	alg	0.8	2.34-5*	8.46-6	6.49-6	40	46	159	153	0.37	0.37	0.29	0.32
stat	alg	1.0	4.05-5	1.14-5	-4.48-7	33	39	166	160	0.48	0.48	0.35	0.38
stat	alg	1.2	2.48-5	7.26-6	4.76-6	28	33	171	166	0.57	0.57	0.41	0.43
stat	alg	1.5	9.13-5	1.15-5	-4.48-6	23	28	176	171	0.68	0.68	0.49	0.52
stat	alg	1.8	1.18-4	8.14-6	-5.42-7	20	24	179	175	0.75	0.75	0.56	0.58
stat	alg	2.0	3.24-5	6.97-6	1.97-6	18	21	181	178	0.79	0.79	0.60	0.62
stat	alg	2.2	1.32-4	1.07-5	-5.79-6	16	20	183	179	0.82	0.82	0.64	0.66
stat	alg	2.5	3.53-5	6.62-6*	1.31-6	14	17	185	182	0.85	0.85	0.68	0.70
stat	alg	2.8	1.41-4	1.15-5	-7.91-6	13	16	186	183	0.88	0.88	0.73	0.74
stat	alg	3.0	1.46-4	1.16-5	-8.39-6	12	15	187	184	0.89	0.89	0.75	0.76
stat	alg	3.2	1.08-4	1.23-5	-8.46-6	11	14	188	185	0.91	0.91	0.77	0.78
stat	alg	3.5	1.72-4	1.16-5	-9.27-6	10	13	189	186	0.92	0.92	0.80	0.81
stat	alg	3.8	1.71-4	1.22-5	-1.02-5	10	12	189	187	0.93	0.93	0.82	0.83
stat	alg	4.0	9.04-5	9.89-6	-5.01-6	9	11	190	188	0.94	0.94	0.83	0.84

Table 19: Optimal scale for e_S , effect of scale in computational errors of *-
*-FD models at uniform nodes. $t_{end} = 3.0, (n_t, n_x) = (4000, 100)$, ratio
 $\Delta t/(\Delta x)^2 = 5/8$.

$Grid$ $Type$	Tr	$FD-$ $Node$	x_{ref}	r_{max}	r_{av}	e_{max}	e_{av}	e_S
stat	exp	unif	2.00*	9e-5	2e-5	4.535-5	1.082-5*	3.326-07
stat	exp	unif	2.02	1e-4	2e-5	4.317-5	1.092-5	8.926-08
stat	exp	unif	2.028	1e-4	2e-5	4.207-5	1.098-5	9.572-09
stat	exp	unif	2.0290	1e-4	2e-5	4.191-5	1.099-5	1.011-09
stat	exp	unif	2.0291*	1e-4	2e-5	4.190-5	1.099-5	*7.829-11
stat	exp	unif	2.0292*	1e-4	2e-5	4.188-5	1.099-5	*-8.507-10
stat	exp	unif	2.0293	1e-4	2e-5	4.187-5	1.099-5	-1.776-09
stat	exp	unif	2.0298	1e-4	2e-5	4.179-5	1.099-5	-5.846-09
stat	exp	unif	2.03*	1e-4	2e-5	4.176-5*	1.100-5	-7.648-09
stat	exp	unif	2.05	1e-4	2e-5	4.251-5	1.093-5	-7.338-08
stat	exp	unif	2.10	7e-5	2e-5	5.754-5	1.156-5	-3.478-07
stat	exp	unif	2.20	9e-5	2e-5	7.128-5	1.230-5	-1.440-06
stat	alg	unif	1.0	2e-5	3e-7	9.521-5	3.424-5	-2.262-06
stat	alg	unif	1.1	8e-5	2e-6	2.319-4	5.007-5	-3.111-05
stat	alg	unif	1.17	7e-5	3e-6	1.846-4	4.816-5	-2.581-05
stat	alg	unif	1.19	9e-5	4e-6	3.537-4	4.631-5	-3.089-05
stat	alg	unif	1.198	3e-4	7e-6	3.647-4	3.090-5	-8.302-06
stat	alg	unif	1.1985	6e-4	9e-6	3.199-4	2.804-5	-1.911-06
stat	alg	unif	1.1986*	7e-4	1e-5	3.026-4	2.731-5	*-5.732-08
stat	alg	unif	1.1987*	8e-4	1e-5	2.792-4	2.653-5	*2.058-06
stat	alg	unif	1.1988	1e-3	1e-5	2.466-4	2.562-5	4.521-06
stat	alg	unif	1.199	3e-3	3e-5	1.253-4	2.336-5	9.991-06
stat	alg	unif	1.2**	3e-4	6e-6	6.969-5*	2.223-5*	1.151-05
dyna	exp	unif	0.2	1e-5	3e-6	2.237-2	1.429-3	-6.764-04
dyna	exp	unif	0.25*	2e-5	4e-6	9.227-4	4.677-5*	-1.420-05
dyna	exp	unif	0.26*	2e-5	4e-6	4.379-4*	4.728-5	-1.989-06
dyna	exp	unif	0.263	2e-5	4e-6	6.397-4	5.500-5	-1.014-06
dyna	exp	unif	0.2636	2e-5	4e-6	6.939-4	5.589-5	-1.395-07
dyna	exp	unif	0.2637*	2e-5	4e-6	7.036-4	5.612-5	*-5.836-08
dyna	exp	unif	0.2638*	2e-5	4e-6	7.132-4	5.635-5	*2.265-08
dyna	exp	unif	0.2639	2e-5	4e-6	7.227-4	5.658-5	9.956-08
dyna	exp	unif	0.264	2e-5	4e-6	7.322-4	5.681-5	1.797-07
dyna	exp	unif	0.27	2e-5	4e-6	1.141-3	6.804-5	2.938-06
dyna	exp	unif	0.3	2e-5	5e-6	1.656-3	7.092-5	1.248-05
dyna	alg	unif	0.2	8e-6	2e-6	3.292-4	6.017-5	-3.174-05
dyna	alg	unif	0.25	1e-5	2e-6	2.291-4	4.197-5	-5.690-06
dyna	alg	unif	0.26	1e-5	2e-6	2.176-4	3.928-5	-2.087-06
dyna	alg	unif	0.263	1e-5	2e-6	2.144-4	3.847-5	-6.789-07
dyna	alg	unif	0.264	1e-5	2e-6	2.133-4	3.821-5	-1.454-07
dyna	alg	unif	0.2641	1e-5	2e-6	2.132-4	3.819-5	-8.917-08
dyna	alg	unif	0.2642*	1e-5	2e-6	2.130-4	3.816-5	*-3.191-08
dyna	alg	unif	0.2643*	1e-5	2e-6	2.129-4	3.814-5	*2.615-08
dyna	alg	unif	0.2644	1e-5	2e-6	2.128-4	3.811-5	8.492-08
dyna	alg	unif	0.265	1e-5	2e-6	2.121-4	3.797-5	4.591-07
dyna	alg	unif	0.27	1e-5	2e-6	2.058-4	3.684-5	1.097-06
dyna	alg	unif	0.3**	1e-5	2e-6	1.752-4*	3.114-5*	1.060-05

Table 20: $x_{ref} = 0.5$. Comparisons among orthogonal nodes : *-dyna-FD vs *-dyna-DM models in unit time span. $t_{end} = 3.0, (n_x, n_t) = (100, 4000), (200, 16000), (400, 64000), \Delta t/(\Delta x)^2 = 5/8$.

Tr	$Grid$ $Type$	$method$	$FD-$ $Nodes$	n_x	x_{ref}	e_{max}	e_{av}	e_s	$t-$ $march$	$Rank$
exp	dyna	FD	unif	100	0.5	9.242-5	9.637-6	2.590-5	0.6	
exp	dyna	FD	unif	200	0.5	2.136-5	2.470-6	6.264-6	4.3	
exp	dyna	FD	*unif	400	0.5	5.298-6	6.517-7*	1.509-6	34.0	E1
exp	dyna	FD	Cheb1	100	0.5	8.426-5	2.970-5	-3.013-6	0.6	
exp	dyna	FD	Cheb1	200	0.5	2.104-5	7.364-6	-7.357-7	4.6	
exp	dyna	FD	Cheb1	400	0.5	5.251-6	1.831-6	-1.785-7	36.7	
exp	dyna	FD	Cheb2	100	0.5	8.064-5	2.884-5	-1.213-6	0.6	
exp	dyna	FD	Cheb2	200	0.5	2.069-5	7.276-6	-5.571-7	4.6	
exp	dyna	FD	*Cheb2	400	0.5	5.215-6*	1.822-6	*-1.603-7	36.7	E2
exp	dyna	DM	spl-unif	100	0.5	7.719-5	3.477-5	3.602-6	3.7	
exp	dyna	DM	spl-unif	200	0.5	1.949-5	8.914-6	5.156-7	75.0	E3
exp	dyna	DM	spl-Cheb1	100	0.5	8.143-5	3.512-5	3.930-6	3.8	
exp	dyna	DM	spl-Cheb1	200	0.5	2.049-5	8.863-6	1.025-6	75.8	
exp	dyna	DM	spl-Cheb2	100	0.5	8.342-5	3.529-5	4.050-6	3.8	
exp	dyna	DM	spl-Cheb2	200	0.5	2.076-5	8.895-6	1.021-6	75.7	
exp	dyna	DM	Spectral	100	0.5	5.506-5	1.394-5	1.196-5	6.5	
exp	dyna	DM	Spectral	200	0.5	1.372-5	3.526-6	3.029-6	187.3	
alg	dyna	FD	unif	100	0.5	8.728-5	1.437-5	3.162-5	0.5	
alg	dyna	FD	unif	200	0.5	2.176-5	3.654-6	8.855-6	3.8	
alg	dyna	FD	*unif	400	0.5	5.433-6*	9.262-7*	2.296-6	30.6	A1
alg	dyna	FD	Cheb1	100	0.5	2.022-4	2.251-5	2.202-5	0.5	
alg	dyna	FD	Cheb1	200	0.5	4.972-5	5.601-6	5.452-6	4.1	
alg	dyna	FD	Cheb1	400	0.5	1.232-5	1.392-6	1.359-6	32.4	
alg	dyna	FD	Cheb2	100	0.5	1.965-4	2.228-5	2.151-5	0.5	
alg	dyna	FD	Cheb2	200	0.5	4.899-5	5.572-6	5.353-6	4.1	
alg	dyna	FD	*Cheb2	400	0.5	1.223-5	1.389-6	1.345-6	32.5	A2
alg	dyna	DM	spl-unif	100	0.5	9.632-5	3.892-5	-1.669-6	3.3	
alg	dyna	DM	spl-unif	200	0.5	2.393-5	9.789-6	*-4.330-7	65.1	A3
alg	dyna	DM	spl-Cheb1	100	0.5	2.059-4	5.512-5	-1.282-5	3.4	
alg	dyna	DM	spl-Cheb1	200	0.5	5.180-5	1.390-5	-3.187-6	65.2	
alg	dyna	DM	spl-Cheb2	100	0.5	2.114-4	5.582-5	-1.324-5	3.4	
alg	dyna	DM	spl-Cheb2	200	0.5	5.249-5	1.400-5	-3.253-6	65.6	
alg	dyna	DM	Spectral	100	0.5	5.447-5	1.379-5	1.118-5	5.7	
alg	dyna	DM	Spectral	200	0.5	1.357-5	3.484-6	2.830-6	160.9	

Table 21: $x_{ref} = 1.5$. Comparisons among orthogonal nodes : *-dyna-FD vs *-dyna-DM models in unit time span. $t_{end} = 3.0, (n_x, n_t) = (100, 4000), (200, 16000), (400, 64000), \Delta t/(\Delta x)^2 = 5/8$.

<i>Tr</i>	<i>Grid Type</i>	<i>method</i>	<i>FD-Nodes</i>	n_x	x_{ref}	e_{max}	e_{av}	e_S	t_{march}	<i>Rank</i>
exp	dyna	FD	unif	100	1.5	7.499-5	1.246-5	3.405-5	0.6	E1
exp	dyna	FD	unif	200	1.5	1.661-5	2.907-6	7.926-6	4.6	
exp	dyna	FD	**unif	400	1.5	2.452-6*	6.340-7*	1.588-6	36.9	
exp	dyna	FD	Cheb1	100	1.5	2.006-4	2.331-5	-2.240-5	0.6	
exp	dyna	FD	Cheb1	200	1.5	6.126-5	7.240-6	-8.331-6	4.8	
exp	dyna	FD	Cheb1	400	1.5	1.589-5	1.845-6	-2.217-6	37.8	
exp	dyna	FD	Cheb2	100	1.5	8.439-5	1.225-5	4.003-6	0.6	E2
exp	dyna	FD	Cheb2	200	1.5	4.814-5	5.780-6	-5.401-6	4.8	
exp	dyna	FD	*Cheb2	400	1.5	1.469-5	1.716-6	-1.959-6	38.0	
exp	dyna	DM	spl-unif	100	1.5	1.249-4	1.613-5	-1.173-5	5.2	
exp	dyna	DM	spl-unif	200	1.5	3.673-5	4.391-6	-3.946-6	132.3	
exp	dyna	DM	spl-Cheb1	100	1.5	1.204-4	1.821-5	-9.529-6	5.2	
exp	dyna	DM	spl-Cheb1	200	1.5	2.292-5	3.811-6	-7.345-7	134.5	E3
exp	dyna	DM	spl-Cheb2	100	1.5	8.246-5	1.420-5	-1.025-6	5.2	
exp	dyna	DM	spl-Cheb2	200	1.5	1.902-5	3.428-6	*1.195-7	132.4	
exp	dyna	DM	Spectral	100	1.5	7.148-5	9.870-6	1.254-6	10.5	
exp	dyna	DM	Spectral	200	1.5	1.561-5	2.243-6	8.484-7	393.8	
alg	dyna	FD	unif	100	1.5	1.605-4	2.302-5	6.656-5	0.5	A1
alg	dyna	FD	unif	200	1.5	2.438-5	4.344-6	1.159-5	4.1	
alg	dyna	FD	**unif	400	1.5	3.689-6*	8.941-7*	2.220-6	32.2	
alg	dyna	FD	Cheb1	100	1.5	1.278-4	2.342-5	-1.020-5	0.5	
alg	dyna	FD	Cheb1	200	1.5	3.738-5	6.317-6	-4.260-6	4.2	
alg	dyna	FD	Cheb1	400	1.5	9.499-6	1.576-6	-1.112-6	33.5	
alg	dyna	FD	Cheb2	100	1.5	7.915-5	1.655-5	7.971-6	0.5	A2
alg	dyna	FD	Cheb2	200	1.5	3.203-5	5.724-6	-2.818-6	4.2	
alg	dyna	FD	*Cheb2	400	1.5	9.016-6	1.522-6	*-9.869-7	33.4	
alg	dyna	DM	spl-unif	100	1.5	1.375-4	2.530-5	-1.688-5	4.9	
alg	dyna	DM	spl-unif	200	1.5	3.619-5	6.624-6	-4.518-6	120.8	
alg	dyna	DM	spl-Cheb1	100	1.5	1.422-4	3.164-5	-1.688-5	4.9	
alg	dyna	DM	spl-Cheb1	200	1.5	3.140-5	7.611-6	-3.047-6	122.5	A3
alg	dyna	DM	spl-Cheb2	100	1.5	1.245-4	3.018-5	-1.199-5	4.9	
alg	dyna	DM	spl-Cheb2	200	1.5	3.040-5	7.544-6	-2.780-6	120.2	
alg	dyna	DM	Spectral	100	1.5	6.483-5	1.009-5	3.296-6	9.5	
alg	dyna	DM	Spectral	200	1.5	1.443-5	2.377-6	1.320-6	347.5	

Table 22: $x_{ref} = 2.5$. Comparisons among orthogonal nodes : *-dyna-FD vs *-dyna-DM models in unit time span. $t_{end} = 3.0, (n_x, n_t) = (100, 4000), (200, 16000), (400, 64000), \Delta t/(\Delta x)^2 = 5/8$.

Tr	$Grid$ $Type$	$method$	$FD-$ $Nodes$	n_x	x_{ref}	e_{max}	e_{av}	e_s	$t-$ $march$	$Rank$
exp	dyna	FD	unif	100	2.5	1.187-4	1.462-5	2.814-5	0.6	
exp	dyna	FD	unif	200	2.5	6.203-5	4.395-6	1.195-5	4.6	
exp	dyna	FD	*unif	400	2.5	1.185-5*	9.967-7*	2.327-6	36.9	E1
exp	dyna	FD	Cheb1	100	2.5	2.573-4	2.089-5	-1.300-5	0.6	
exp	dyna	FD	Cheb1	200	2.5	1.183-4	9.791-6	-1.163-5	4.8	
exp	dyna	FD	Cheb1	400	2.5	3.674-5	2.959-6	-4.031-6	37.8	
exp	dyna	FD	Cheb2	100	2.5	1.632-4	2.489-5	3.362-5	0.6	
exp	dyna	FD	Cheb2	200	2.5	4.969-5	4.063-6	-8.913-7	4.8	
exp	dyna	FD	*Cheb2	400	2.5	3.013-5	2.446-6	-3.098-6	37.6	E2
exp	dyna	DM	spl-unif	100	2.5	1.877-4	1.457-5	-1.742-5	5.6	
exp	dyna	DM	spl-unif	200	2.5	7.837-5	4.957-6	-8.599-6	138.0	
exp	dyna	DM	spl-Cheb1	100	2.5	2.544-4	2.458-5	-2.544-5	5.6	
exp	dyna	DM	spl-Cheb1	200	2.5	5.072-5	4.966-6	-4.334-6	138.9	
exp	dyna	DM	spl-Cheb2	100	2.5	1.080-4	1.431-5	-4.377-6	5.7	
exp	dyna	DM	spl-Cheb2	200	2.5	2.112-5	3.179-6	*-2.253-7	140.1	E3
exp	dyna	DM	Spectral	100	2.5				failed	
exp	dyna	DM	Spectral	200	2.5	2.665-5	2.438-6	-9.935-7	392.2	
alg	dyna	FD	unif	100	2.5	2.332-4	2.434-5	5.857-5	0.5	
alg	dyna	FD	unif	200	2.5	4.764-5	5.376-6	1.197-5	4.1	
alg	dyna	FD	*unif	400	2.5	9.218-6*	1.162-6*	2.470-6	32.3	A1
alg	dyna	FD	Cheb1	100	2.5	2.973-4	2.937-5	-3.211-5	0.5	
alg	dyna	FD	Cheb1	200	2.5	8.636-5	8.338-6	-1.054-5	4.2	
alg	dyna	FD	Cheb1	400	2.5	2.252-5	2.172-6	-2.844-6	33.5	
alg	dyna	FD	Cheb2	100	2.5	1.200-4	1.753-5	-4.983-8	0.5	
alg	dyna	FD	Cheb2	200	2.5	6.202-5	6.324-6	-6.202-6	4.2	
alg	dyna	FD	*Cheb2	400	2.5	2.015-5	1.965-6	-2.426-6	33.6	A2
alg	dyna	DM	spl-unif	100	2.5	1.136-4	1.645-5	-7.334-6	5.4	
alg	dyna	DM	spl-unif	200	2.5	4.635-5	5.447-6	-4.877-6	128.5	
alg	dyna	DM	spl-Cheb1	100	2.5	1.616-4	2.457-5	-1.481-5	5.4	
alg	dyna	DM	spl-Cheb1	200	2.5	3.736-5	5.969-6	-3.090-6	131.1	
alg	dyna	DM	spl-Cheb2	100	2.5	1.276-4	2.159-5	-8.550-6	5.4	
alg	dyna	DM	spl-Cheb2	200	2.5	3.120-5	5.438-6	*-2.005-6	131.2	A3
alg	dyna	DM	Spectral	100	2.5	7.148-5	8.501-6	8.903-7	10.9	
alg	dyna	DM	Spectral	200	2.5	1.694-5	2.067-6	4.440-7	360.5	

Table 23: Performance of FD method in the finite interval in long-term simulation. $t_{end} = 12.0$. Focused on two ratios $\Delta t/(\Delta x)^2 = (10/320000)/(2/400)^2 = 10/8$, and $(10/160000)/(2/800)^2 = 10$.

<i>Case</i>	n_t	n_x	<i>Ratio</i>	<i>Tr</i>	x_{ref}	<i>Grid Type</i>	<i>Node</i>	e_{max}	e_{av}	e_S	t_{-march}
1	40000	400	10/8	alg	1.0	stat	unif	6.395-6	2.902-6	2.058-6	16.7
2	80000	400		alg	1.0	stat	unif	6.616-6	2.833-6	8.679-7	33.5
3	160000	400		alg	1.0	stat	unif	7.073-6	2.820-6	2.722-7	67.4
4	320000	400		alg	1.0	stat	unif	7.301-6	2.816-6	-2.530-8	134.5
5	640000	400		alg	1.0	stat	unif	7.416-6	2.816-6	-1.739-7	267.6
6	10000	200	10/8	exp	1.0	stat	unif	1.917-5	9.280-6	1.023-5	2.5
7	20000	200		exp	1.0	stat	unif	1.905-5	5.149-6	5.301-6	4.9
8	40000	200		exp	1.0	stat	unif	1.976-5	3.084-6	2.836-6	9.6
9	80000	200		exp	1.0	stat	unif	2.012-5	2.051-6	1.603-6	19.8
10	160000	200		exp	1.0	stat	unif	2.030-5	1.587-6	9.871-7	39.5
11	40000	400	10/8	exp	1.0	stat	unif	5.010-6	2.370-6↓	2.463-6	19.7
12	80000	400		exp	1.0	stat	unif	4.793-6	1.338-6↓	1.230-6	39.2
13	160000	400		exp	1.0	stat	unif	4.959-6	8.216-7↓	6.135-7	78.4
14	320000	400		exp	1.0	stat	unif	5.042-6	5.634-7↓	3.053-7	*157.0
15	640000	400		exp	1.0	stat	unif	5.083-6	4.347-7↓	1.512-7	305.1
16	320000	400	10/8	exp	0.5	stat	unif	6.383-4	1.502-5	2.256-6	148.0
17	320000	400	10/8	exp	0.8	stat	unif	4.521-5	6.095-7	4.406-7	149.1
18	320000	400	10/8	exp	1.0	stat	unif	5.042-6	5.634-7*	3.036-7	*152.8
19	320000	400	10/8	exp	1.5	stat	unif	1.357-6	7.630-7	-4.657-8	158.4
20	320000	400	10/8	exp	2.0	stat	unif	2.421-6	9.815-7	-3.035-7	163.0
21	320000	400	10/8	exp	2.5	stat	unif	3.781-6	1.212-6	-6.670-7	163.6
22	160000	800	10	exp	0.5	stat	unif	4.589-4	4.113-6	1.256-6	141.7
23	160000	800	10	exp	0.8	stat	unif	1.171-5	5.993-7	7.406-7	148.3
24	160000	800	10	exp	1.0	stat	unif	1.275-6	5.977-7*	6.036-7	*156.3
25	160000	800	10	exp	1.5	stat	unif	1.499-6	5.983-7	3.657-7	162.4
26	160000	800	10	exp	2.0	stat	unif	1.786-6	6.166-7	2.035-7	165.2
27	160000	800	10	exp	2.5	stat	unif	2.140-6	6.375-7	7.670-8	163.3
28	320000	400	10/8	exp	0.5	dyna	unif	6.381-4	1.488-5	3.431-6	164.4
29	320000	400	10/8	exp	0.8	dyna	unif	4.529-5	5.867-7	1.470-6	170.7
30	320000	400	10/8	exp	1.0	dyna	unif	5.064-6	5.348-7*	1.583-6	*173.5
31	320000	400	10/8	exp	1.5	dyna	unif	2.362-6	7.178-7	1.859-6	178.9
32	320000	400	10/8	exp	2.0	dyna	unif	4.316-6	8.743-7	2.077-6	186.7
33	320000	400	10/8	exp	2.5	dyna	unif	6.336-6	1.004-6	2.237-6	187.7
30	320000	400	10/8	exp	1.0	dyna	unif	5.064-6	5.348-7*	1.583-6	*173.5
34	320000	400	10/8	exp	1.0	dyna	Cheb2	6.920-6	2.876-6	-2.000-6	187.7
35	320000	400	10/8	alg	1.0	dyna	unif	7.109-6	1.746-6	2.649-6	154.4
36	320000	400	10/8	alg	1.0	dyna	Cheb2	1.707-5	2.066-6	-2.879-8	165.2

Table 24: Performance of FD vs DM methods in long-term simulations. $t_{end} = 12.0$, $x_{ref} = 1.0$. $(n_x, n_t) = (100, 40000), (200, 160000), (500, 1000000)$, all with small time steps, $\Delta t/(\Delta x)^2 = 5/8$.

Case	n_x	Tr	Grid Type	Meth	Node	e_{max}	e_{av}	e_S	t_{solve}	t_{march}	Rank
1	100	exp	stat	FD	unif	8.39-5	7.08-6	6.98-6	0.2	5.1	1
	200	exp	stat	FD	unif	2.03-5	1.59-6*	9.87-7	1.8	39.5	
	100	exp	dyna	FD	unif	8.50-5	1.49-5	2.94-5	0.9	5.5	
	100	exp	dyna	FD	hyper1	4.34-4	2.47-5	-1.19-5	1.3	5.8	
	100	exp	dyna	FD	hyper2	1.50-4	6.35-5	-5.38-5	1.3	6.1	
	100	exp	dyna	FD	Cheb1	1.15-4	4.30-5	-3.48-5	1.2	5.9	
	100	exp	dyna	FD	Cheb2	1.12-4	3.97-5	-2.89-5	1.2	6.0	
2	200	exp	dyna	FD	unif	2.04-5	3.13-6*	6.30-6	6.7	43.3	2
3	200	exp	dyna	FD	hyper1	8.93-5	6.16-6	-3.18-6	9.5	45.4	
4	200	exp	dyna	FD	hyper2	3.74-5	1.59-5	-1.38-5	9.7	47.4	
5	200	exp	dyna	FD	Cheb1	2.82-5	1.07-5	-8.87-6	8.7	46.5	
6	200	exp	dyna	FD	Cheb2	2.77-5	1.04-5	-8.30-6	8.6	46.5	
7	100	alg	stat	FD	unif	1.03-4	3.62-5	1.52-5	0.3	4.3	4
	200	alg	stat	FD	unif	2.71-5	1.01-5*	1.53-6	1.7	34.7	
	100	alg	dyna	FD	unif	1.12-4	3.15-5	3.92-5	0.9	4.9	
	100	alg	dyna	FD	hyper1	3.30-4	6.12-5	5.16-5	1.3	5.2	
	100	alg	dyna	FD	hyper2	4.66-4	4.66-5	-4.21-6	1.3	5.3	
	100	alg	dyna	FD	Cheb1	2.79-4	3.08-5	-2.20-6	1.2	5.2	
	100	alg	dyna	FD	Cheb2	2.70-4	3.00-5	-1.39-6	1.2	5.2	
8	200	alg	dyna	FD	unif	2.81-5	7.87-6	1.02-5	6.6	38.4	3
9	200	alg	dyna	FD	hyper1	8.21-5	1.53-5	1.30-5	9.6	41.1	
10	200	alg	dyna	FD	hyper2	1.17-4	1.16-5	-9.22-7	9.6	41.2	
11	200	alg	dyna	FD	Cheb1	6.89-5	7.63-6	-6.74-7	8.6	40.7	
12	200	alg	dyna	FD	Cheb2	6.79-5	7.56-6*	-6.74-7	8.5	40.7	
100	exp	dyna	DM	spl-unif	7.23-5	3.52-5	-2.25-5	31.9	40.2		
100	exp	dyna	DM	spl-Cheb1	1.00-4	4.35-5	-3.04-5	32.5	40.8		
100	exp	dyna	DM	spl-Cheb2	1.03-4	4.35-5	-2.99-5	32.5	40.8	DM1	
100	exp	dyna	DM	Spectral	1.63-5	4.48-6	1.69-6	65.1	73.2		
100	alg	dyna	DM	spl-unif	1.01-4	4.66-5	-4.00-5	29.4	35.6		
100	alg	dyna	DM	spl-Cheb1	2.34-4	8.07-5	-8.20-5	30.0	36.0		
100	alg	dyna	DM	spl-Cheb2	2.40-4	8.22-5	-8.37-5	29.9	36.0		
100	alg	dyna	DM	Spectral	1.52-5	4.05-6	2.06-6	57.5	63.5		
13	200	exp	dyna	DM	spl-unif	1.82-5	9.15-6*	-6.15-6	752.7		850.8
14	200	exp	dyna	DM	spl-Cheb1	2.52-5	1.09-5	-7.38-6	762.3	859.7	DM2
15	200	exp	dyna	DM	spl-Cheb2	2.55-5	1.09-5	-7.39-6	765.0	867.6	
16	200	exp	dyna	DM	Spectral	3.82-6	1.07-6	5.56-7	2114.2	*2211.7	
17	200	alg	dyna	DM	spl-unif	2.63-5	1.19-5*	-1.03-5	663.8	733.0	
18	200	alg	dyna	DM	spl-Cheb1	5.89-5	2.05-5	-2.05-5	671.0	737.8	dyna-1
19	200	alg	dyna	DM	spl-Cheb2	5.97-5	2.06-5	-2.08-5	678.4	747.9	
20	200	alg	dyna	DM	Spectral	3.68-6	9.91-7	5.71-7	1829.6	*1896.2	
21	500	exp	dyna	FD	unif	3.27-6*	4.45-7*	8.92-7	221.4	1497.8	
22	500	exp	dyna	FD	Cheb2	4.42-6	1.68-6	-1.39-6	312.1	1667.4	dyna-2
23	500	alg	dyna	FD	unif	4.49-6	1.23-6	1.59-6	216.2	1366.7	
24	500	alg	dyna	FD	Cheb2	1.08-5	1.21-6	-1.14-7	299.9	1472.3	

Table 25: FD vs DM in finite interval models on fine grid in long-term simulations and relatively coarse time meshes, with $t_{end} = 12.0$, $\Delta t/(\Delta x)^2 = 125/8$. Performance of cases are judged in the balance of least average error and less cpu-time. The optimal case (4b) here is slightly better than the *tough* references in Table 8.

<i>Case</i>	x_{ref}	<i>Tr</i>	<i>Grid Type</i>	<i>Meth</i>	<i>FD-Nodes</i>	$\frac{n_t}{100}$	$\frac{n_x}{100}$	e_{max}	e_{av}	e_s	t_{march}	<i>Rank</i>
	1.5	exp	stat	FD	unif	16	1	1.20-4	5.17-5	4.80-5	0.3	
	1.5	exp	stat	FD	unif	64	2	3.30-5	1.34-5	1.08-5	2.3	
	1.5	exp	stat	FD	unif	256	4	8.68-6	3.42-6	2.55-6	18.0	
	1.5	exp	stat	FD	unif	1024	8	2.22-6	8.61-7	6.20-7	103.5	2
	1.5	alg	stat	FD	unif	16	1	1.15-4	4.10-5	5.18-5	0.3	
	1.5	alg	stat	FD	unif	64	2	4.53-5	1.31-5	6.21-6	2.1	
	1.5	alg	stat	FD	unif	256	4	9.76-6	2.83-6	2.20-6	15.4	
	1.5	alg	stat	FD	unif	1024	8	2.92-6	7.96-7*	*3.58-7	88.9	1
	1.5	exp	dyna	FD	unif	400	5	1.31-5	4.28-6	3.32-6	26.8	
	1.5	exp	dyna	FD	Cheb2	400	5	1.77-5	5.64-6	9.76-7	28.7	
	1.5	alg	dyna	FD	unif	400	5	1.11-5	3.86-6	3.93-6	23.8	
	1.5	alg	dyna	FD	Cheb2	400	5	1.50-5	4.97-6	2.17-6	25.2	
1a	0.5	exp	stat	FD	unif	1600	10	3.93-4	2.75-6↓	1.10-6↓	181.6	A5
1b	1.5	exp	stat	FD	unif	1600	10	1.43-6	5.52-7↓	3.95-7↓	197.3	B2
1c	2.5	exp	stat	FD	unif	1600	10	1.85-6	5.46-7↓	1.52-7↓	206.9	C2
2a	0.5	exp	dyna	FD	unif	1600	10	3.93-4	3.28-6↓	1.21-6↓	203.1	A6
2b	1.5	exp	dyna	FD	unif	1600	10	3.08-6	8.99-7↓	6.96-7↓	221.1	B4
2c	2.5	exp	dyna	FD	unif	1600	10	2.41-6	6.49-7↓	6.25-7↓	226.3	C3
3a	0.5	exp	dyna	FD	Cheb2	1600	10	1.87-5	2.50-6↓	1.80-6	218.7	A4
3b	1.5	exp	dyna	FD	Cheb2	1600	10	4.78-6	1.18-6↓	4.92-8	234.7	B6
3c	2.5	exp	dyna	FD	Cheb2	1600	10	6.50-6	1.15-6↓	-4.03-7	234.3	C6
4a	0.5	alg	stat	FD	unif	1600	10	2.80-6	6.90-7	8.48-7↓	166.6	A1
4b	1.5	alg	stat	FD	unif	1600	10	1.67-6	4.65-7*	3.10-7↓	*173.7	B1
4c	2.5	alg	stat	FD	unif	1600	10	2.63-6	4.74-7	8.68-9↓	174.2	C1
5a	0.5	alg	dyna	FD	unif	1600	10	3.38-6	8.37-7	1.41-6↓	187.8	A2
5b	1.5	alg	dyna	FD	unif	1600	10	2.82-6	8.92-7	8.18-7↓	195.2	B3
5c	2.5	alg	dyna	FD	unif	1600	10	2.45-6	7.13-7	7.10-7↓	200.6	C4
6a	0.5	alg	dyna	FD	Cheb2	1600	10	6.88-6	1.25-6↓	8.10-7↓	198.0	A3
6b	1.5	alg	dyna	FD	Cheb2	1600	10	4.06-6	1.15-6↓	3.32-7↓	204.5	B5
6c	2.5	alg	dyna	FD	Cheb2	1600	10	5.12-6	1.08-6↓	-6.21-8↓	206.3	C5
7b	1.5	exp	dyna	DM	slp-unif	400	1	7.67-5	2.08-5	-2.30-5	45.5	
8b	1.5	exp	dyna	DM	slp-unif	1600	1	6.59-5	1.80-5	-2.43-5	182.1	
9b	1.5	exp	dyna	DM	slp-unif	400	2	3.17-5	8.34-6	-4.87-6	267.9	
10b	1.5	exp	dyna	DM	slp-Cheb2	400	1	5.24-5	1.82-5	-1.27-5	46.1	
11b	1.5	exp	dyna	DM	slp-Cheb2	1600	1	4.12-5	1.56-5	-1.40-5	184.2	
12b	1.5	exp	dyna	DM	slp-Cheb2	400	2	2.39-5	7.09-6	-1.65-6	270.3	
13b	1.5	exp	dyna	DM	Spectral	400	1	2.07-5	4.66-6	-4.51-7	88.3	
14b	1.5	exp	dyna	DM	Spectral	800	1	1.32-5	2.91-6	-1.33-6	176.1	
15b	1.5	exp	dyna	DM	Spectral	400	2	1.54-5	3.63-6	1.61-6	740.9	

Table 26: Performance of FD on $[0, +\infty)$ (truncated to $[0, 10]$) vs *-stat-FD models on $[-1, 1]$ at various scales in unit time span, $t_{end} = 3.0$. All cases assume (sub)uniform nodes, and relatively small time steps with $\Delta t/(\Delta x)^2 = 5/8$ on the finite interval.

<i>Case</i>	x_{ref}	<i>Tr</i>	<i>Grid Type</i>	n_t	n_x	e_{max}	e_{av}	e_S	t_{march}	<i>Rank</i>
1a	0.5	exp	stat	4000	100	9.066-05	6.652-06	1.617-05	0.6	A1
1b	0.5	exp	stat	16000	200	2.114-05	1.697-06	3.701-06	4.3	B1
1c	0.5	exp	stat	64000	400	5.267-06	4.392-07	8.812-07	31.7	C1
1d*	0.5	exp	stat	256000	800	1.316-06	1.116-07*	2.149-07	269.8	D1
2a	1.5	exp	stat	4000	100	2.113-05	8.198-06	5.670-06	0.6	A2
2b	1.5	exp	stat	16000	200	8.420-06	2.322-06	5.199-07	4.7	B2
2c	1.5	exp	stat	64000	400	2.292-06	6.065-07	3.968-08	34.8	C2
2d*	1.5	exp	stat	256000	800	6.217-07*	1.549-07	*-1.092-10	292.5	D2
3a	2.5	exp	stat	4000	100	1.052-04	1.437-05	-4.510-06	0.6	A3
3b	2.5	exp	stat	16000	200	2.723-05	3.789-06	-2.109-06	4.7	B3
3c	2.5	exp	stat	64000	400	6.837-06	9.409-07	-5.376-07	35.0	C3
3c	2.5	exp	stat	256000	800	1.640-06	2.317-07	-1.308-07	276.6	D3
4a	0.5	alg	stat	4000	100	8.517-05	3.698-05	4.426-06	0.5	A5
4b	0.5	alg	stat	16000	200	2.406-05	1.055-05	-7.306-07	3.8	B5
4c	0.5	alg	stat	64000	400	6.679-06	2.921-06	-6.793-07	30.4	C7
4d	0.5	alg	stat	256000	800	2.000-06	8.057-07	-3.117-07	242.6	D8
5a	1.5	alg	stat	4000	100	4.287-04	4.216-05	-2.701-05	0.5	A6
5b	1.5	alg	stat	16000	200	9.396-05	1.149-05	-9.029-06	3.8	B6
5c	1.5	alg	stat	64000	400	1.583-05	2.826-06	-2.251-06	30.3	C6
5d	1.5	alg	stat	256000	800	2.466-06	6.285-07	-4.451-07	252.1	D6
6a	2.5	alg	stat	4000	100	7.943-05	1.771-05	1.409-05	0.5	A4
6b	2.5	alg	stat	16000	200	2.915-05	6.670-06	-1.732-06	4.1	B4
6c	2.5	alg	stat	64000	400	1.056-05	2.077-06	-1.302-06	30.0	C4
6d	2.5	alg	stat	256000	800	3.653-06	6.134-07	-5.193-07	255.9	D5
7a			stat	4000	100	initial	$n_W = 2$		failed	
7b			stat	16000	200	4.338-04	1.451-05	-1.238-04	3.6	B7
7c			stat	64000	400	8.909-05	3.178-06	-2.245-05	28.7	C8
7d			stat	256000	800	1.767-05	6.973-07	-4.255-06	229.1	D7
8a			dyna	4000	100	initial	$n_W = 2$		failed	
8b			dyna	16000	200	6.517-04	2.048-05	-1.674-04	4.1	B8
8c			dyna	64000	400	5.163-05	2.404-06	-1.283-05	33.1	C5
8d			dyna	256000	800	5.857-06	3.704-07	1.887-06	259.0	D4

Table 27: Performance of FD in $[0, +\infty)$ (truncated to $[0, 20]$) vs *-stat-FD models in long-term simulation, $t_{end} = 12.0$. All parameters on finite interval models are the same as in Table 26, $\Delta t/(\Delta x)^2 = 5/8$.

<i>Case</i>	<i>x_{ref}</i>	<i>Tr</i>	<i>Grid Type</i>	<i>n_t</i>	<i>n_x</i>	<i>e_{max}</i>	<i>e_{av}</i>	<i>e_S</i>	<i>t- march</i>	<i>Rank</i>
1a	0.5	exp	stat	40000	100	4.396-04	1.064-04	7.869-05	4.8	A6
1b	0.5	exp	stat	160000	200	5.744-04	4.607-05	1.703-05	36.2	B6
1c	0.5	exp	stat	640000	400	6.383-04	1.516-05	2.745-06	296.0	C7
1d	0.5	exp	stat	2560000	800	4.588-04	4.190-06	3.878-07	2356.2	D7
2a	1.5	exp	stat	40000	100	2.058-05	7.949-06	4.616-06	5.2	A1
2b	1.5	exp	stat	160000	200	5.235-06	2.328-06	-5.350-08	40.0	B1
2c	1.5	exp	stat	640000	400	1.319-06	6.499-07	-1.598-07	326.1	C1
2d	1.5	exp	stat	2560000	800	3.310-07	1.706-07	-5.794-08	2558.5	D1
3a	2.5	exp	stat	40000	100	2.762-05	1.409-05	-3.183-06	5.2	A2
3b	2.5	exp	stat	160000	200	1.213-05	4.236-06	-2.311-06	42.2	B2
3c	2.5	exp	stat	640000	400	3.462-06	1.120-06	-7.254-07	333.1	C2
3c	2.5	exp	stat	2560000	800	9.353-07	2.869-07	-1.975-07	2641.3	D2
4a	0.5	alg	stat	40000	100	2.876-04	7.077-05	-1.992-05	4.1	A5
4b	0.5	alg	stat	160000	200	8.946-05	1.865-05	-3.752-06	33.5	B5
4c	0.5	alg	stat	640000	400	1.792-05	4.752-06	-1.500-07	259.0	C5
4d	0.5	alg	stat	2560000	800	4.478-06	1.150-06	1.766-07	2084.4	D5
5a	1.5	alg	stat	40000	100	1.148-04	3.911-05	5.220-06	4.5	A3
5b	1.5	alg	stat	160000	200	4.435-05	1.250-05	-5.471-06	34.5	B4
5c	1.5	alg	stat	640000	400	7.642-06	2.663-06	-7.155-07	279.8	C4
5d	1.5	alg	stat	2560000	800	2.752-06	7.513-07	-3.723-07	2219.4	D4
6a	2.5	alg	stat	40000	100	1.843-04	4.672-05	-6.795-06	4.6	A4
6b	2.5	alg	stat	160000	200	2.936-05	8.343-06	1.355-07	35.7	B3
6c	2.5	alg	stat	640000	400	8.582-06	2.480-06	-9.991-07	278.7	C3
6d	2.5	alg	stat	2560000	800	3.098-06	7.111-07	-4.708-07	2232.7	D3
7a			stat	40000	100	initial	$n_w = 1$		failed	
7b			stat	160000	200	initial	$n_w = 2$		failed	
7c			stat	640000	400	2.256-04	1.021-05	-1.217-04	269.5	C6
7d			stat	2560000	800	4.519-05	2.135-06	-2.345-05	2135.6	D6
8a			dyna	40000	100	initial	$n_w = 1$		failed	
8b			dyna	160000	200	initial	$n_w = 2$		failed	
8c			dyna	640000	400	initial	$n_w = 4$		failed	
8d			dyna	2560000	800	initial	$n_w = 8$		failed	