Article:
Naumann, A, Ruprecht, D orcid.org/0000-0003-1904-2473 and Wensch, J (2018) Toward transient finite element simulation of thermal deformation of machine tools in real-time. Computational Mechanics, 62 (5). pp. 929-942. ISSN 0178-7675

https://doi.org/10.1007/s00466-018-1540-6

Computational Mechanics manuscript No.
(will be inserted by the editor)

Toward transient finite element simulation of thermal deformation of machine tools in real-time

Andreas Naumann · Daniel Ruprecht · Joerg Wensch

the date of receipt and acceptance should be inserted later

Abstract Finite element models without simplifying assumptions can accurately describe the spatial and temporal distribution of heat in machine tools as well as the resulting deformation. In principle, this allows to correct for displacements of the Tool Centre Point and enables high precision manufacturing. However, the computational cost of FE models and restriction to generic algorithms in commercial tools like ANSYS prevents their operational use since simulations have to run faster than real-time. For the case where heat diffusion is slow compared to machine movement, we introduce a tailored implicit-explicit multi-rate time stepping method of higher order based on spectral deferred corrections. Using the open-source FEM library DUNE, we show that fully coupled simulations of the temperature field are possible in real-time for a machine consisting of a stock sliding up and down on rails attached to a stand.

Keywords machine tool · thermal error · real-time simulation · numerical time-stepping · spectral deferred corrections

This work was funded by the German Research Foundation as part of the CRC/TR 96.

Andreas Naumann · Joerg Wensch
Institut for Scientific Computing
Technische Universität Dresden
Dresden
Germany
E-mail: Andreas.Naumann@tu-dresden.de

Daniel Ruprecht
School of Mechanical Engineering
University of Leeds
Woodhouse Lane
LS2 9JT
Leeds
United Kingdom

1 Introduction

Machine tools that are capable of correcting for displacements of the Tool Centre Point (TCP) caused by thermal expansion are a promising approach for high precision manufacturing (other approaches involve, e.g., design modification or thermal-error control) [21]. Most machine tools these days are "intelligent" and employ sensors to measure temperature. Compensating for thermal errors requires knowledge of the deviation from the machine's reference temperature.

Since the moving parts of a machine result in strongly position and time dependent heat sources and deformations [24, 25], this knowledge should ideally include spatial and temporal variations to account for position-dependent heating and transient effects. Since sensors can only provide data at isolated points, computational models are required to complement measured data and obtain accurate temperature distributions. Obviously, to allow for the correction of thermal errors during operations, any model to be used for online error compensation has to run faster than real-time in the sense that the "look-ahead factor" satisfies

$$\eta = \frac{\text{simulated time}}{\text{wall-clock time}} > 1.$$

In operational use, the heat distribution would be predicted by the transient FEM solved online, probably informed by sensor measurements. At selected time intervals, the resulting deformation field would be derived from the computed temperature field. The predicted dislocation of the TCP would then be corrected for. Running online FEM to predict the heat distribution also allows to consider varying external factors like, e.g., environment temperature.

The faster the FEM models runs, the larger the look-ahead factor η becomes and the further into the future the simulation can "see". If, e.g., we simulate the machine's deformation over 10 s and this simulation requires 5 s to run, we achieve a look-ahead factor of $\eta = 2$. A larger look-ahead factor gives more time to apply the necessary corrections. Furthermore, a very large look-ahead factor would allow to run the model several times, either to address uncertainties through, e.g., an ensemble approach or to compute the necessary correction within an optimal control framework.

Finite element models (FEM) are derived from first principles and can thus provide a reliable and detailed description of heat transfer and diffusion, even though accurate specification of boundary conditions can be a challenge [22]. Accurate transient finite element models are very useful as they can provide spatially and temporally resolved temperature fields for machines with complex designs and geometries [25,18]. In contrast to empirical approaches [3,23], the parameters in FEM are physical quantities that can, at least theoretically, be measured. Since reduced models are typically machine-specific, their derivation also comes with a high cost in terms of person hours. In contrast, the mesh for FEM can be generated automatically, e.g. from CAD files, even for machines with complex geometries.

The disadvantage of FEM is their high computational cost, which is why often reduced models are employed, sacrificing accuracy or generality for speed. Running full time-dependent FEM is considered too computationally expensive to be possible in real-time: "application of the original FE-models without any simplifications [...] for model-based control-integrated correction is very time-consuming and thus impractical" [15]. Despite only resolving one machine part and employing a time-averaged heat source instead of a full coupling, Galant et al. report a computation time of around 5 hours to simulate a milling machine with 16,626 degrees-of-freedom over 16 hours using ANSYS (corresponding to $\eta = 3.2$). To the best of the authors' knowledge, there are no reports of simulations solving in real-time the fully coupled transient FEM problem for a machine with moving parts without simplifications. Recent review papers also make no mention of such efforts [21,25,20]. A combination of finite differences and FEM, called FDEM, has been proposed that reduces computational effort but still relies on the use of macro elements to reduce the size of the solved system [24]. With respect to FDM and FEM, in a review from 2017, Cao et al. state that "[...], due to the low efficiency, the computational models were rarely used in online thermal error compensation" [7], mentioning only approaches that rely on steady-state FE models [9,10].

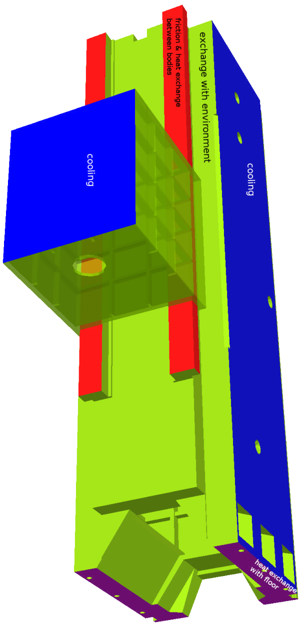
A key reason is probably that while widely used commercial proprietary software like ANSYS [1] is easy to use, this simplicity comes with a performance penalty and restriction to generic numerical methods that do not consider the special structure of the problem. To solve the fully coupled problem in ANSYS, e.g., only implicit Euler is applicable [27]. While implicit Euler is a robust and widely used time stepping method, it is only first order accurate and does not take into account the different time scales involved, leaving room for substantial efficiency gains by using more tailored algorithms of higher order.

Contributions We demonstrate that accurate faster than real-time simulations with a full transient FE model with 16,626 degrees of freedom are possible by implementing a tailored higher order multi-rate time stepping method in the open-source finite element library DUNE [4,14,30]. While open-source FEM libraries are typically more difficult to use than commercial packages, they are flexible and offer efficient implementations of spatial discretisations and solvers and can be tailored to specific problems.

Our time stepping method is based on multi-rate spectral deferred corrections (MRSDC) [11,5,19,6,13], focussing on the case where the movement of the machine is fast compared to diffusive heat transport and simulated time equals multiple complete machine cycles. It combines implicit treatment of heat diffusion over larger time steps with explicit integration of the machine movement over smaller steps. This avoids stability issues from the diffusive term, maintains accuracy for the fast dynamics induced by machine movement and avoids the need to reassemble a Jacobian in each step. We demonstrate that the new method can substantially improve computational efficiency. For a look-ahead factor of $\eta = 10$, implicit Euler provides time discretisation errors of about 20% which is probably too inaccurate to compute useful information about the machine deformation. In contrast, for the same value of η, MRSDC is about one order of magnitude more accurate, yielding an error of about 3%. For a smaller look-ahead factor of $\eta = 2$, implicit Euler can provide errors of about 1% while MRSDC is again about an order of magnitude more accurate, providing an error below 0.1%.

2 Description of the problem

Figure 1 shows the configuration of the machine. The stand and rail are fixed and the corresponding computational domain is labeled Ω_f. The stock moves up and down along the rail (red surface in Figure 1) and we refer to this part of the domain as Ω_m. The equations


cooling
friction & heat exchange between bodies
exchange with environment
cooling
heat exchange with floor

parameter		value	unit
ρ		7200	$\frac{\mathrm{kg}}{m^3}$
C_p		460	$\frac{J}{\mathrm{kg}\cdot K}$
ν		50	$\frac{W}{mK}$
E		$2.1 \cdot 10^{11}$	$\frac{N}{m^2}$
ν_P		0.3	-
α_e		$1.2 \cdot 10^{-5}$	$\frac{1}{K}$
α		50	$\frac{W}{m^2K}$
α_i	1	10	
	2	100	
	3	100	
T_i	1	24.0…24.5	$°C$
	2	22.0	
	3	20.0	

Fig. 1 Fixed stand and stock moving along the rails. Colors indicate different types of boundary conditions: cooling at the right side of the stand and the back of the stock (blue), heat exchange with the floor at the bottom (violet), heat exchange between stand/stock plus heat generated by friction (red) and heat exchange with the environment (green) at all other boundaries.

modelling diffusion of heat within the two parts read

$$\rho C_p \partial_t T_{\mathrm{f}} = \nu \Delta T_{\mathrm{f}} \text{ in } \Omega_{\mathrm{f}} \tag{1a}$$
$$\rho C_p \partial_t T_{\mathrm{m}} = \nu \Delta T_{\mathrm{m}} \text{ in } \Omega_{\mathrm{m}}(t). \tag{1b}$$

Both geometries are coupled through the heat flux at the moving common boundary segment $\Gamma_{\mathrm{R}}(t) = \Gamma_{\mathrm{m}}(t) \cap \Gamma_{\mathrm{f}}$ at the rail. In Figure 1, $\Gamma_{\mathrm{R}}(t)$ corresponds to the part of the top of the rail covered by the stock at time t. At Γ_R we have heat exchange between stock and rail and heat generation due to friction. For the sake of simplicity, we assume thermal isolation at the rest of the rail. Put together, we obtain boundary conditions

$$\nu \nabla T_{\mathrm{f}} \cdot \vec{n} = 0 \text{ on } \Gamma_{\mathrm{f,rail}} \setminus \Gamma_{\mathrm{m}}(t) \tag{2a}$$
$$\nu \nabla T_{\mathrm{m}} \cdot \vec{n} = \alpha \left(T_{\mathrm{f}} - T_{\mathrm{m}}\right) + \frac{\eta(t)}{2} \text{ on } \Gamma_{\mathrm{R}}(t) \tag{2b}$$
$$\nu \nabla T_{\mathrm{f}} \cdot \vec{n} = \alpha \left(T_{\mathrm{m}} - T_{\mathrm{f}}\right) + \frac{\eta(t)}{2} \text{ on } \Gamma_{\mathrm{R}}(t). \tag{2c}$$

We consider here that case where both the moving and fixed part are made of the same material, so that $\nu_{\mathrm{fix}} = \nu_{\mathrm{mov}} = \nu$, see the table in Figure 1. But using different values for conductivity would be straightforward.

Both domains are also thermally coupled to the surrounding air and a cooling equipment, modelled by Robin boundary conditions at the static pieces of the machine

$$\nu \nabla T_{\mathrm{f}} \cdot \vec{n} = \alpha_{\mathrm{i}} \left(T_{\mathrm{i}} - T_{\mathrm{f}}\right) \text{ on } \Gamma_{\mathrm{f,i}} \tag{3a}$$
$$\nu \nabla T_{\mathrm{m}} \cdot \vec{n} = \alpha_{\mathrm{i}} \left(T_{\mathrm{i}} - T_{\mathrm{m}}\right) \text{ on } \Gamma_{\mathrm{m,i}} \tag{3b}$$

for $i = 1, 2, 3$. Here, index $i = 1$ represents the boundary where heat is exchanged with the environment (green in Figure 1), index $i = 2$ the boundary where cooling is applied (blue in Figure 1) and finally $i = 3$ heat exchange with the floor (violet in Figure 1). Each boundary uses a different value for T_{i} and α_{i}. The environmental temperature is assumed to be equal to 24 °C at the floor with a slight increase of 0.5 °C over the 2 m distance to the top of the stand, modelling a sunlit workshop on a warm day.

Both domains are meshed independently and equations (1) are discretized using linear finite elements. Meshes Ω_{f} and Ω_{m} have different basis and test function spaces V_{f} and V_{m}. Multiplying equations (1) with the corresponding test functions, integrating each domain separately and inserting the boundary conditions yields

$$\begin{aligned}\rho C_p \int_{\Omega_{\mathrm{f}}} \partial_t T_{\mathrm{f}} \varphi_{\mathrm{f}} dx = -\nu \int_{\Omega_{\mathrm{f}}} \nabla T_{\mathrm{f}} \nabla \varphi_{\mathrm{f}} dx \\ + \int_{\Gamma_{\mathrm{f,env}}} \alpha_{\mathrm{f}} (T_{\mathrm{env}} - T_{\mathrm{f}}) \varphi_{\mathrm{f}} dS \\ + \int_{\Gamma_{\mathrm{R}}(t)} \left(\alpha (T_{\mathrm{m}} - T_{\mathrm{f}}) + \frac{\eta(t)}{2} \right) \varphi_{\mathrm{f}} dS\end{aligned} \tag{4a}$$

$$\begin{aligned}\rho C_p \int_{\Omega_{\mathrm{m}}} \partial_t T_{\mathrm{m}} \varphi_{\mathrm{m}} dx = -\nu \int_{\Omega_{\mathrm{m}}} \nabla T_{\mathrm{m}} \nabla \varphi_{\mathrm{m}} dx \\ + \int_{\Gamma_{\mathrm{m,env}}} \alpha_{\mathrm{m}} (T_{\mathrm{env}} - T_{\mathrm{m}}) \varphi_{\mathrm{m}} dS \\ + \int_{\Gamma_{\mathrm{R}}(t)} \left(\alpha (T_{\mathrm{f}} - T_{\mathrm{m}}) + \frac{\eta(t)}{2} \right) \varphi_{\mathrm{m}} dS\end{aligned} \tag{4b}$$

for every test function φ_{f} and φ_{m} respectively. By representing the solutions $T_{\mathrm{f}}(x) = \sum_k \vec{T}_{\mathrm{f},k}(t) \varphi_{\mathrm{f},k}(x)$ and $T_{\mathrm{m}}(x) = \sum_k \vec{T}_{\mathrm{m},k} \varphi_{\mathrm{m},k}(x)$ in basis functions on the corresponding mesh we can write the continuous equations (4a) and (4b) in their discrete forms

$$\begin{aligned}\mathbf{M}_{\mathrm{f}} \partial_t \vec{T}_{\mathrm{f}} =& \mathbf{A}_{\mathrm{f}} \vec{T}_{\mathrm{f}} + \mathbf{B}_{\mathrm{f,env}} \vec{T}_{\mathrm{f}} + \vec{b}_{\mathrm{f,env}} \\ & + \mathbf{M}_{\mathbf{B}\mathrm{f}}(t) \vec{T}_{\mathrm{f}} + \mathbf{C}_{\mathbf{B}\mathrm{m,f}}(t) \vec{T}_{\mathrm{m}} \\ & + \vec{b}_{\mathrm{f}}(t)\end{aligned} \tag{5a}$$

$$\begin{aligned}\mathbf{M}_{\mathrm{m}} \partial_t \vec{T}_{\mathrm{m}} =& \mathbf{A}_{\mathrm{m}} \vec{T}_{\mathrm{m}} + \mathbf{B}_{\mathrm{m,env}} \vec{T}_{\mathrm{m}} + \vec{b}_{\mathrm{m,env}} \\ & + \mathbf{M}_{\mathbf{B}\mathrm{m}} \vec{T}_{\mathrm{m}} + \mathbf{C}_{\mathbf{B}\mathrm{f,m}}(t) \vec{T}_{\mathrm{f}} \\ & + \vec{b}_{\mathrm{m}}(t)\,.\end{aligned} \tag{5b}$$

To avoid duplication, we use a generic subscript X instead of “m” and “f”, for “move” and “fixed” respectively, when expressions are identical on both parts. In both equations we have the standard mass matrix $\mathbf{M}_X = \int_{\Omega_X} \varphi_X \varphi_X dx$ and the discrete Laplacian $\mathbf{A}_X = -\nu \int_{\Omega_X} \nabla\varphi_X \cdot \nabla\varphi_X dx$ for every test function φ_X. We have split the contributions from the environment in two parts. The first part

$$\mathbf{B}_{X,\mathrm{env}} = -\int_{\Gamma_{X,\mathrm{env}}} \alpha_X \varphi_X \varphi_X dS \tag{6}$$

depends on the machine temperature while the second part

$$\overrightarrow{b}_{\mathrm{X,\ env}} = \int_{\Gamma_{\mathrm{X,env}}} \alpha_{\mathrm{X}} T_{\mathrm{env}} \varphi_{\mathrm{X}} dS \tag{7}$$

does not. The time dependent source term

$$\overrightarrow{b}_{\mathrm{X}}(t) = \int_{\Gamma_R(t)} \frac{\eta(t)}{2} \varphi_{\mathrm{X}} dS \tag{8}$$

models heat generation through friction [27]. We split the term $\mathbf{M}_X(t)\overrightarrow{T}_X + \mathbf{C}_{\mathbf{B}Y,X}\overrightarrow{T}_Y$ modelling heat exchange between stand and stock in two terms. The coefficients of the first part

$$\mathbf{M}_X = -\int_{\Gamma_R(t)} \alpha \varphi_X \varphi_X dS \tag{9}$$

are similiar to the matrix from the Robin boundary condition, whereas the matrix of the second part

$$\mathbf{C}_{\mathbf{B}Y,X} = \int_{\Gamma_R(t)} \alpha \varphi_Y \varphi_X dS \tag{10}$$

contains basis functions from both domains. This is the term which couples the temperature fields of both machine components. Since both parts are meshed independently, the meshes do not match at the interface. Therefore, we have to compute the intersections of both meshes using existing methods for grid coupling in DUNE [4, 14] to evaluate the boundary integrals in (10).

Now, we combine both equations in (5) into one coupled system

$$\begin{aligned}\begin{pmatrix}\mathbf{M}_{\mathrm{f}} & 0\\ 0 & \mathbf{M}_{\mathrm{m}}\end{pmatrix}\partial_t \overrightarrow{T} = {}& \begin{pmatrix}\mathbf{A}_{\mathrm{f}} + \mathbf{B}_{\mathrm{f,env}} & 0\\ 0 & \mathbf{A}_{\mathrm{m}} + \mathbf{B}_{\mathrm{m,env}}\end{pmatrix}\overrightarrow{T}\\ &+ \begin{pmatrix}\overrightarrow{b}_{\mathrm{f,env}}\\ \overrightarrow{b}_{\mathrm{m,env}}\end{pmatrix} + \begin{pmatrix}\mathbf{M}_{\mathbf{B}\mathrm{f}}(t) & \mathbf{C}_{\mathbf{B}\mathrm{m,f}}(t)\\ \mathbf{C}_{\mathbf{B}\mathrm{m,f}}^T(t) & \mathbf{M}_{\mathbf{B}\mathrm{m}}\end{pmatrix}\overrightarrow{T}\\ &+ \begin{pmatrix}\overrightarrow{b}_{\mathrm{f}}(t)\\ \overrightarrow{b}_{\mathrm{m}}(t)\end{pmatrix}\end{aligned} \tag{11}$$

by introducing $\overrightarrow{T} = \begin{pmatrix}\overrightarrow{T}_{\mathrm{f}}\\ \overrightarrow{T}_{\mathrm{m}}\end{pmatrix}$. We also replace the mass matrices $\mathbf{M}_{\mathrm{f}}$ and $\mathbf{M}_{\mathrm{m}}$ by their row sum-lumped version [2].

In preparation for the introduction of the multi-rate time stepping in the next section, we split the right hand side function into the following parts

$$\overrightarrow{f}^I(\overrightarrow{T}) = \begin{pmatrix}\mathbf{A}_{\mathrm{f}} + \mathbf{B}_{\mathrm{f,env}} & 0\\ 0 & \mathbf{A}_{\mathrm{m}} + \mathbf{B}_{\mathrm{m,env}}\end{pmatrix}\overrightarrow{T} \tag{12}$$

$$\overrightarrow{g}(t) = \begin{pmatrix}\overrightarrow{b}_{\mathrm{f,env}}\\ \overrightarrow{b}_{\mathrm{m,env}}\end{pmatrix} \tag{13}$$

$$\overrightarrow{f}^E(\overrightarrow{T}, t) = \begin{pmatrix}\mathbf{M}_{\mathbf{B}\mathrm{f}}(t) & \mathbf{C}_{\mathbf{B}\mathrm{m,f}}(t)\\ \mathbf{C}_{\mathbf{B}\mathrm{m,f}}^T(t) & \mathbf{M}_{\mathbf{B}\mathrm{m}}\end{pmatrix}\overrightarrow{T} + \begin{pmatrix}\overrightarrow{b}_{\mathrm{f}}(t)\\ \overrightarrow{b}_{\mathrm{m}}(t)\end{pmatrix}. \tag{14}$$

With this notation (11) can compactly be written as

$$\mathbf{M}\partial_t \overrightarrow{T}(t) = \overrightarrow{F}(\overrightarrow{T}(t), t) = \overrightarrow{f}^I(\overrightarrow{T}(t)) + \overrightarrow{f}^E(\overrightarrow{T}(t), t) + \overrightarrow{g}(t). \tag{15}$$

Since we consider the regime where heat diffusion is slow compared to machine movement, $\overrightarrow{f}^I$ represents a slow process. In contrast, $\overrightarrow{f}^E$ represents the fast coupling process and generation of heat from friction. Evaluating $\overrightarrow{f}^E$ requires detecting the intersection of finite elements at the interface between the two components, which can be expensive. Lastly, $\overrightarrow{g}(t)$ models heat exchanges with the environment, floor and cooling, which are also slow relative to the movement of the machine.

This splitting is motivated by the following time scale argument. Heat in the machine dissipates towards a stationary state on a time scale $\tau_c = \frac{\rho C_p L^2}{\nu}$, where L is a characteristic length. The second time scale τ_m is given by the time it takes the stand to travel along the rail once. One limiting case would be $\tau_c/\tau_m \to 0$. Here, machine movement is infinitely fast compared to heat diffusion which would result in some aggregate effective heat source on the rail without resolving stock movement. The other limiting case is $\tau_c/\tau_m \to \infty$, where heat across the machine adjusts to a new steady state instantaneously after each infinitesimal movement of the stock without transients.

For the machine studied here, the edge length $L = 0.5\,\mathrm{m}$ leads to $\tau_c = 16{,}560\,\mathrm{s}$. In contrast, the material parameters in Table 1 result in a diffusive time scale of $\tau_m = 12\,\mathrm{s}$. Therefore, we have a clear scale separation in the sense that $\tau_c \gg \tau_m$ which we exploit in our algorithm.

Deformations. The stationary deformation $\vec{u}$ is given by the equations of linear elasticity

$$\mu \Delta \vec{u} + (\lambda + \mu)\nabla\left(\nabla \cdot \vec{u}\right) = \vec{f}_{\text{th}} + \vec{f}_{\text{mech}}\,, \tag{16}$$

where the thermal stress

$$\vec{f}_{\text{th}} = \alpha(3\lambda + 2\mu)\nabla \cdot I[T(t) - T(0)] \tag{17}$$

is due to deviation from the reference temperature [16, 8]. We use Lame‘s parameters $\mu = \frac{E}{2(1+\nu_P)}$ and $\lambda = \frac{E\nu_P}{(1+\nu)(1-2\nu_P)}$ but specified the Young‘s modulus E and Poisson‘s ratio ν_P in the table in Figure 1. We focus here only on thermally induced deformations and ignore mechanical loads, so that $\vec{f}_{\text{mech}} = 0$. This is in line with what is typically done in the literature [15, 24] and most papers do not even mention the mechanical part. A detailed investigation of the role of u_{mech} would be interesting but is beyond the scope of this paper.

3 Numerical time stepping method

In this section, we present a time stepping algorithm with a problem-specific multi-rate splitting based on spectral deferred corrections [11] that will reduce solution times significantly compared to a standard implicit Euler method.

The term $\vec{f}^{\,I}$ models diffusion of heat. Resolving it accurately requires a time step $\Delta t = \mathcal{O}(\Delta x)$ while stability for an explicit integrator requires $\Delta t = \mathcal{O}(\Delta x^2)$. To resolve all geometrical features of the machine the mesh has many small elements with diameters of the order of 2×10^{-4} m whereas the stand is 2 m high and has a 0.5 m by 0.5 m base. Given the values for ν, ρ, C_p in Figure 1, an explicit integrator would require a time step

$$\Delta t \leq \frac{\rho C_p \Delta x^2}{\nu} = 0.0027\,\text{s} \tag{18}$$

for stability, which is orders of magnitude too small to be efficient. Therefore, f^I is treated implicitly with a larger time step. The term $\vec{g}$ is independent of $\vec{T}$ and models heat exchange with the environment which is a slow process. Therefore, we use the same large time step as for $\vec{f}^{\,I}$.

In contrast, for $\vec{f}^{\,E}$, modelling the movement of parts of the machine, we require that $\Delta t = \mathcal{O}(\frac{\Delta x}{v})$ (v being the speed of the machine). Otherwise, the stock moves across multiple mesh cells in one time step, creating a “stroboscope effect” and highly unrealistic temperature distributions [28]. There is thus no benefit integrating $\vec{f}^{\,E}$ implicitly because taking a large time step is impossible anyway. Furthermore, implicit treatment of this term leads to a time-dependent Jacobian and a potentially large number of evaluations, each of which would require detecting intersections. To avoid both issues, we integrate $\vec{f}^{\,E}$ explicitly but with a smaller time step.

Finally, to achieve better computational efficiency, we want our time stepping method to be at least second order accurate. Derivation of both implicit-explicit and multi-rate method of higher order is challenging and we employ the spectral deferred corrections framework for this purpose.

When using a finer spatial mesh, computing the intersections becomes cheaper relative to solving the full problem. Therefore, we expect the cost of $\vec{f}^{\,E}$ to go down compared to $\vec{f}^{\,I}$, making the splitting approach more efficient. However, this will also depend on the precise stability properties of the method which we have not yet studied in detail. A corresponding analysis is beyond the scope of this paper and left for future work.

Single-rate spectral deferred correction

Before discussing the multi-rate spectral deferred correction (SDC) algorithm, we first describe its single-rate variant. Consider the initial value problem (15) over one time step $[t_n, t_{n+1}]$. Let

$$t_n \leq \tau_1 < \ldots < \tau_M \leq t_{n+1}$$

denote a set of quadrature nodes within the time step. We denote the distances between nodes by $\Delta\tau_m = \tau_m - \tau_{m-1}$ for $m = 2, \ldots, M$ and $\Delta\tau_1 = \tau_1 - t_n$. The analytical solution of (15) satisfies the integral equations

$$\mathbf{M}\vec{T}(\tau_m) = \mathbf{M}\vec{T}(t_n) + \int_{t_n}^{\tau_m} \vec{F}(\vec{T}(s), s)\, ds \tag{19}$$

for $m = 1, \ldots, M$. We approximate the integral using a quadrature rule, resulting in the discrete approximations

$$\mathbf{M}\vec{T}_m = \mathbf{M}\vec{T}(t_n) + \sum_{j=1}^{M} q_{m,j}\vec{F}(\vec{T}_j, \tau_j) \tag{20}$$

of (19) with $\vec{T}_m \approx \vec{T}(\tau_m)$. The quadrature weights $q_{m,j}$ are given as integrals over Lagrange polynomials [29]. This approach is known as collocation and the unknowns $\vec{T}_m$ correspond to the stages of a fully implicit Runge-Kutta method with Butcher tableau [17, Theorem 7.7]. Theoretically, these can be computed using a Newton-Raphson method to solve the M coupled nonlinear equations (20) but the large size of the nonlinear system makes this approach impractical for systems with a large number of degrees-of-freedom, in particular semi-discrete partial differential equations. Subtracting equations (19) for m and $m-1$ yields the "node-to-node" variants of the integral equations

$$\mathbf{M}\vec{T}(\tau_m) = \mathbf{M}\vec{T}(\tau_{m-1}) + \int_{\tau_{m-1}}^{\tau_m} \vec{F}(\vec{T}(s), s)\, ds. \tag{21}$$

Instead, spectral deferred corrections employ an iterative procedure which avoids assembly of the full system. Each iteration can be computed by a "sweep" through the quadrature nodes with a low order method. Semi-implicit SDC (SISDC) [26] starts with an initial prediction step using IMEX-Euler to generate approximate values $\vec{T}^0_m$ from

$$\begin{aligned}\mathbf{M}\vec{T}^0_m = \mathbf{M}\vec{T}^0_{m-1} + \Delta\tau_m \left(\vec{f}^I(\vec{T}^0_m) + \vec{g}(\tau_m)\right)\\ +\Delta\tau_m \vec{f}^E(\vec{T}^0_{m-1}, \tau_{m-1})\end{aligned} \tag{22}$$

for $m = 1, \ldots, M$ with $\vec{T}^0_0 = \vec{T}(t_n)$. This provides a first order accurate approximation of $\vec{T}$ at the quadrature nodes. Then, to increase the order, SISDC proceeds with the following iterative correction

$$\begin{aligned}\mathbf{M}\vec{T}^{k+1}_m = \mathbf{M}\vec{T}^{k+1}_{m-1} + \Delta\tau_m \left(\vec{f}^I(\vec{T}^{k+1}_m) - \vec{f}^I(\vec{T}^k_m)\right)\\ + \Delta\tau_m \left(\vec{f}^E(\vec{T}^{k+1}_{m-1}, \tau_{m-1}) - \vec{f}^E(\vec{T}^k_{m-1}, \tau_{m-1})\right)\\ + I^m_{m-1}.\end{aligned} \tag{23}$$

with

$$I^m_{m-1} := \sum_{j=1}^{M} s_{m,j} \vec{F}(\vec{T}^k_j, \tau_j) \approx \int_{\tau_{m-1}}^{\tau_m} \vec{F}(\vec{T}(s), s)\, ds. \tag{24}$$

The weights are given by $s_{m,j} := q_{m,j} - q_{m-1,j}$ for $m = 2, \ldots, M$ and $s_{1,j} := q_{1,j}$. Note that since the source term $\vec{g}$ does not change with k, $\vec{g}(\tau_m) - \vec{g}(\tau_m)$ cancels out but it is considered in the correction steps through $\vec{F}(\vec{T}^k_j, \tau_j)$ in the quadrature term I^m_{m-1}.

For $k \to \infty$, if the iteration converges and $\vec{T}^{k+1}_m - \vec{T}^k_m \to 0$ for $m = 1, \ldots, M$, Equation (23) reduces to

$$\mathbf{M}\vec{T}_m = \mathbf{M}\vec{T}_{m-1} + I^m_{m-1} \tag{25}$$

which is the discrete counterpart of (21). Applying this equation recursively to retrieve the zero-to-node variant shows that the $\vec{T}^{k+1}_m$ converge to the solutions $\vec{T}_m$ of (20). However, the appeal of SDC stems from the fact that it is not necessary to fully solve the collocation problem. It can be shown [29] that, if the time step is small enough, each iteration reduces the residual

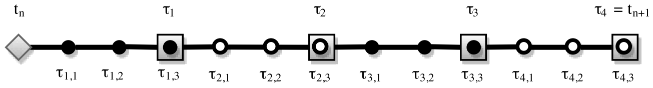


Fig. 2 Standard quadrature nodes τ_m, $m = 1, \ldots, M$ (grey squares) and embedded quadrature nodes $\tau_{m,p}$, $p = 1, \ldots, P$ (black and white circles) within a time step $[t_n, t_{n+1}]$ in multi-rate spectral deferred correction (MRSDC) for $M = 4$ standard nodes and $P = 3$ embedded nodes. We use no-left equidistant nodes, that is t_n is not a standard quadrature nodes and $\tau_1 = \tau_{1,3}$ is part of the first set of embedded nodes $(\tau_{1,j})_{j=1,\ldots P}$ in $[t_n, \tau_1]$, but not an embedded node in $[\tau_1, \tau_2]$.

$$r^k := \max_{m=1,\ldots,M} \left\| \mathbf{M}\vec{T}^k_m - \mathbf{M}\vec{T}(t_n) - \sum_{j=1}^{m} I^m_{m-1} \right\| \tag{26}$$

by a factor proportional to Δt. Therefore, each iteration increases the formal order of the method by one, up to the order of the underlying quadrature rule which depends on M and the chosen type of nodes. Thus, by adjusting the runtime parameter K and M, SISDC allows to generate a split scheme of arbitrary order.

Multi-rate spectral deferred correction

Multi-rate SDC (MRSDC) has been first introduced by Bourlioux, Layton and Minion [5]. In MRSDC, a set of *embedded* quadrature nodes $\tau_{m,p}$, $p = 1, \ldots, P$, is introduced in between each pair $[\tau_{m-1}, \tau_m]$ of *standard* quadrature nodes as illustrated in Figure 2. Therefore, we have a total of $M \times P$ nodes

$$t_n \le \tau_{1,1} < \ldots < \tau_{1,P} \le \tau_1 < \ldots < \tau_{M,P} \le \tau_M \le t_{n+1}. \tag{27}$$

For simplicity, we assume here that the rightmost quadrature node always coincides with the endpoint of the interval so that $\tau_M = t_{n+1}$ and $\tau_{m,P} = \tau_m$. Furthermore, we use equidistant quadrature nodes where t_n is not a standard node (that is, $t_n < \tau_1$) and τ_{m-1} is not a quadrature node for the embedded nodes $\tau_{m,p}$ (that is, $\tau_{m-1} < \tau_{m,1}$). While equidistant nodes limit the formal order of the quadrature rule to the number of nodes (instead of, e.g., twice the number of nodes for Gauss-Legendre quadrature), it significantly improves SDC's convergence in the very stiff limit [32]. The fast changing term $\vec{f}^E$ is approximated by a cumulative sum of the embedded nodes (that is, with small steps) while the slowly changing terms $\vec{f}^I$ and $\vec{g}$ are approximated only at the standard nodes.

Analogously to the equations (21) at the standard nodes, the integral equations at the embedded nodes read

$$\mathbf{M}\overrightarrow{T}(\tau_{m,p}) = \mathbf{M}\overrightarrow{T}(\tau_{m,p-1}) + \int_{\tau_{m,p-1}}^{\tau_{m,p}} \overrightarrow{F}(\overrightarrow{T}(s), s)\ ds. \tag{28}$$

In addition to the approximations at the standard nodes $\overrightarrow{T}_m \approx \overrightarrow{T}(\tau_m)$ as in single-rate SDC we now also consider approximations $\overrightarrow{T}_{m,p} \approx \overrightarrow{T}(\tau_{m,p})$ of the solution $\overrightarrow{T}$ at the embedded nodes. Then, we approximate the integrals with the following quadrature rules

$$I_{m-1}^{m} \approx \int_{\tau_{m-1}}^{\tau_m} \overrightarrow{F}(\overrightarrow{T}(s), s)\ ds \tag{29}$$

with

$$\begin{aligned} I_{m-1}^{m} :=& \sum_{j=1}^{M} s_{m,j} \left(f^I(\overrightarrow{T}_j) + g(\tau_j) \right) \\ &+ \sum_{p=1}^{P} \hat{s}_{m,p} f^E(\overrightarrow{T}_{m,p}, \tau_{m,p}) \end{aligned} \tag{30}$$

and

$$I_{m,p-1}^{p} \approx \int_{\tau_{m,p-1}}^{\tau_{m,p}} \overrightarrow{F}(\overrightarrow{T}(s), s)\ ds \tag{31}$$

with

$$\begin{aligned} I_{m,p-1}^{p} :=& \sum_{j=1}^{M} \tilde{s}_{m,p,j} \left(f^I(\overrightarrow{T}_j) + g(\tau_j) \right) \\ &+ \sum_{q=1}^{P} s_{m,p,q} f^E(\overrightarrow{T}_{m,q}, \tau_{m,q}). \end{aligned} \tag{32}$$

The quadrature weights are defined as follows: let $l_m(s)$ denote the Lagrange polynomials with respect to the standard nodes and $l_{m,p}(s)$ the Lagrange polynomials with respect to one set of embedded nodes, that is

$$l_m(\tau_j) = \delta_{mj}, \quad m, j = 1, \ldots, M \tag{33}$$

and

$$\begin{aligned} l_{m,p}(\tau_{m,q}) = \delta_{pq},\ m = 1, \ldots, M;\ p = 1, \ldots, P; \\ q = 1, \ldots, P \end{aligned} \tag{34}$$

with δ being the Kronecker Delta. Then, the weights are defined as

$$s_{m,j} := \int_{\tau_{m-1}}^{\tau_m} l_j(s)\ ds \tag{35a}$$

$$\hat{s}_{m,p} := \int_{\tau_{m-1}}^{\tau_m} l_{m,p}(s)\ ds \tag{35b}$$

$$\tilde{s}_{m,p,j} := \int_{\tau_{m,p-1}}^{\tau_{m,p}} l_j(s)\ ds \tag{35c}$$

$$s_{m,p,q} := \int_{\tau_{m,p-1}}^{\tau_{m,p}} l_{m,q}(s)\ ds. \tag{35d}$$

Integral boundaries	Position of function values	
	Standard	Embedded
Standard	$s_{m,j}$	$\hat{s}_{m,q}$
Embedded	$\tilde{s}_{m,p,j}$	$s_{m,p,q}$

Table 1 Quadrature weights for integrals between standard or embedded nodes depending on whether approximate function values are given at standard or embedded nodes.

Thus, the weights $s_{m,j}$ and $\hat{s}_{m,p,j}$ approximate integrals between standard nodes while $\tilde{s}_{m,p}$ and $s_{m,p,q}$ approximate integrals between embedded nodes, see Table 1.

Now we approximate the continuous integral equations (21) and (28) with their discrete counterparts

$$\mathbf{M}\overrightarrow{T}_m = \mathbf{M}\overrightarrow{T}_{m-1} + I_{m-1}^{m} \tag{36}$$

and

$$\mathbf{M}\overrightarrow{T}_{m,p} = \mathbf{M}\overrightarrow{T}_{m,p-1} + I_{m,p-1}^{p}. \tag{37}$$

Note that the integral approximations are consistent in the sense that

$$I_{m-1}^{m} = \sum_{p=1}^{P} I_{m,p-1}^{p} \tag{38}$$

because

$$s_{m,j} = \sum_{p=1}^{P} \tilde{s}_{m,p,j} \quad \text{and} \quad \hat{s}_{m,p} = \sum_{q=1}^{P} s_{m,p,q}. \tag{39}$$

Just as for the single-rate case, we consider the residual (26) at the standard nodes. In theory, we could solve the $(M+1) \times P$ nonlinear equations (36) and (37) directly for the $\overrightarrow{T}_m$ and $\overrightarrow{T}_{m,p}$. However, solving such a large system is impractical so we again rely on an iterative approximation.

We start by generating a first order accurate approximation at all nodes (standard and embedded) by computing the predictor step shown in Algorithm 1. For every standard step $[\tau_{m-1}, \tau_m]$, we first compute one large implicit Euler step to generate an estimated final value $\overrightarrow{T}_m^*$ and compute $f_m^* = f^I(\overrightarrow{T}_m^*)$. Then, we compute a series of small steps using explicit Euler in $\overrightarrow{f}^E$, going from $\tau_{m-1} = \tau_{m,0}$ to $\tau_{m,P} = \tau_m$. The implicit term remains fixed to f_m^* throughout. Finally, we use the final results of the series of small steps $\overrightarrow{T}_{m,P}^0$ as $\overrightarrow{T}_m^0$, that is as initial value for the next embedded step. This then provides the initial value $\overrightarrow{T}_m^0$ for the next interval from τ_m to τ_{m+1} where we start the procedure again with a large implicit step. As we show later, the predictor step provides a first order accurate approximation.

Algorithm 1: Multi-rate SDC prediction step.

```
input  : $\vec{T}(t_n)$
output: $\vec{T}^0_m$ and $\vec{T}^0_{m,p}$ for $m = 1, \dots, M$ and
        $P = 1, \dots, P$.
$\vec{T}^0_0 \leftarrow \vec{T}(t_n)$
for $m = 1, M$ do
    /* Implicit step over $[\tau_{m-1}, \tau_m]$.                */
    Solve $\mathbf{M}\vec{T}^*_m = \mathbf{M}\vec{T}^0_{m-1} + \Delta t_m \left(f^I(\vec{T}^*_m) + g(\tau_m)\right)$
    $f^*_m \leftarrow f^I(\vec{T}^*_m) + g(\tau_m)$
    /* Set starting value at $\tau_{m-1}$.                        */
    $\vec{T}^0_{m,0} \leftarrow \vec{T}^0_{m-1}$
    /* Sweep through embedded nodes $\tau_{m,p}$ with
       explicit Euler.                                            */
    for $p = 1, P$ do
        $\mathbf{M}\vec{T}^0_{m,p} = \mathbf{M}\vec{T}^0_{m,p-1} +$
          $\Delta t_{m,p} \left(f^*_m + f^E(\vec{T}^0_{m,p-1}, \tau_{m,p-1})\right)$
    end
    /* Update value at $\tau_m$ by overwriting with
       final value from embedded sweep (since
       $\tau_{m,P} = \tau_m$).                                    */
    $\vec{T}^0_m \leftarrow \vec{T}^0_{m,P}$
end
```

The order is then increased using the iteration shown in Algorithm 2. It proceeds similarly to the predictor step by combining a single large implicit step in $\vec{f}^I$ over $[\tau_{m-1}, \tau_m]$ with P many small explicit steps for the embedded nodes. Through numerical examples, we will demonstrate the following properties.

(i) **Convergence to collocation solution:** the residual (26) decreases geometrically proportional to Δt and approximately at the same rate as for single-rate SDC.
(ii) **Order of accuracy:** each iteration increases the formal order by one, up to the order of the approximations of the integral $\min\{M, P\}$.
(iii) **Computational efficiency:** multi-rate SDC reduces solution times while maintaining the same accuracy as single-rate SDC or implicit Euler.
(iv) **Smooth temperature profiles:** the smaller time step for the coupling in MRSDC leads to a smoother temperature profile than implicit Euler which results in more realistic deformations since those depend on the temperature gradient.

Properties (i) and (ii) are demonstrated for a two dimensional problem of reduced complexity while (iii) and (iv) are demonstrated for the fully coupled 3D machine.

Algorithm 2: Multi-rate SDC correction sweep.

```
input  : $\vec{T}(t_n)$ and $\vec{T}^k_m$, $\vec{T}^k_{m,p}$ for $m = 1, \dots, M$,
         $p = 1, \dots, P$.
output: updated values $\vec{T}^{k+1}_m$, $\vec{T}^{k+1}_{m,p}$
/* Update the integral terms                                      */
Update $I^m_{m-1}$, $m = 1, \dots, M$ according to (29)
Update $I^p_{m-1,p-1}$, $m = 1, \dots, M$; $p = 1, \dots, P$
  according to (31)
/* Value at beginning of time step $\tau_0 = t_n$ is
   brought forward from previous step and remains
   the same for all iterations $k$.                               */
$\vec{T}^{k+1}_0 \leftarrow \vec{T}(t_n)$
for $m = 1, M$ do
    /* Implicit correction step over $[\tau_{m-1}, \tau_m]$.
       Note that the $g(\tau_m)$ term cancels out but is
       included in $I^m_{m-1}$.                                   */
    Solve $\mathbf{M}\vec{T}^*_m =$
      $\mathbf{M}\vec{T}^{k+1}_{m-1} + \Delta t_m \left(f^I(\vec{T}^*_m) - f^I(\vec{T}^k_m)\right) + I^m_{m-1}$
    $f^*_m \leftarrow f^I(\vec{T}^*_m) - f^I(\vec{T}^k_m)$
    /* Set starting value at $\tau_{m-1}$.                        */
    $\vec{T}^{k+1}_{m,0} \leftarrow \vec{T}^{k+1}_{m-1}$
    for $p = 1, P$ do
        /* Sweep through embedded nodes $\tau_{m,p}$ with
           explicit Euler.                                        */
        $\mathbf{M}\vec{T}^{k+1}_{m,p} = \mathbf{M}\vec{T}^{k+1}_{m,p-1} + \Delta t_{m,p} f^*_m$
          $+ \Delta t_{m,p} f^E(\vec{T}^{k+1}_{m,p-1}, \tau_{m,p-1})$
          $- \Delta t_{m,p} f^E(\vec{T}^k_{m,p-1}, \tau_{m,p-1})$
          $+ I^p_{m,p-1}$
    end
    /* Update value at $\tau_m$ by overwriting with
       final value from embedded sweep (since
       $\tau_{m,P} = \tau_m$).                                    */
    $\vec{T}^{k+1}_m \leftarrow \vec{T}^{k+1}_{m,P}$
end
```

4 Convergence to collocation solution and formal order of accuracy

We demonstrate the theoretical properties (i) and (ii) of the method for a simplified 2D version of the full problem that is cheap to solve and allows to easily run simulations for a wide range of parameters. The configuration is sketched in Figure 3. In this scenario, the stand is a rectangle with variable temperature while the stock is represented as a smaller rectangle of constant temperature T_0, gliding left and right. We neglect the rails and prescribe the heat flux $\nu \nabla T_{\text{fix}} = \alpha(T_0 - T)$ at the intersection. The stock moves horizontally with velocity $v = -0.1\,\mathrm{m\,s^{-1}}$. At the remaining boundaries, we assume thermal isolation and apply a zero flux condition. The stand is discretized with bilinear finite elements. In the similar way as for the 3D problem we obtain the equation

$$\mathbf{M}\partial_t \vec{T} = (\nu \mathbf{A} + \alpha \mathbf{M_B}(t))\,\vec{T} + \alpha \mathbf{C_B}(t) \mathbb{1} T_0\,. \tag{40}$$

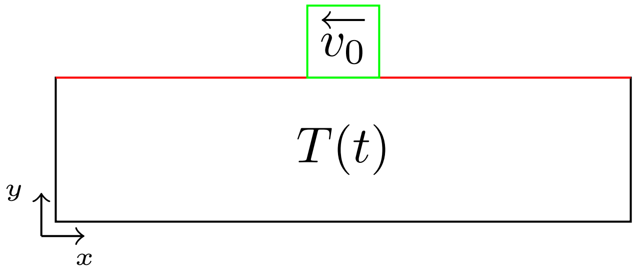


Fig. 3 The 2D domain with the red boundary line at the top. This red line represents the moving flux with reference temperature v_0 and the remaining black lines correspond to zero flux.

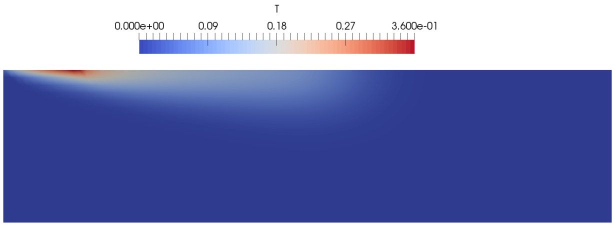


Fig. 4 The temperature field after 20 seconds, i.e. the source square reached the left corner. The temperature tail from the center to the left boundary is clearly visible

The matrices $\mathbf{A}$, $\mathbf{M}$, $\mathbf{M_B}$ and $\mathbf{C_B}$ are analogous to the full problem, but we dropped the subscript "fix". Our splitting is now straightforward. The slow implicit part f^I and the fast part f^E are

$$\vec{f}^{\,I}(\vec{T}) = \nu \mathbf{A} \vec{T} \tag{41a}$$

$$\vec{f}^{\,E}(\vec{T}, t) = \alpha \mathbf{M_B}(t) \vec{T} + \alpha \mathbf{C_B}(t) \mathbb{1} T_0 \tag{41b}$$

whereas $g(t) = 0$ due to the fact the we neglect the thermal exchange with the environment. Figure 4 shows the temperature field after $t = 20$s when the stock is located at the left side. Following the movement of the stock, the temperature increases at the boundary from the center to the left corner with heat dissipating slowly into the stand.

To demonstrate (i), Figure 5 shows the residual (26) plotted against the iteration index k for three different time steps. As a guide to the eye, lines proportional to Δt^k are shown. Results from SDC with $M = 5$ nodes (dashed lines) and MRSDC with $(M, P) = (5, 8)$ nodes (solid lines) are shown. Residuals are nearly identical for both methods with only small differences for the largest step size $\frac{t_e}{5}$. For both methods, the residuals decay proportional to Δt^k so that smaller time steps lead to faster convergence. Eventually, both methods reproduce the collocation solution up to machine precision. Note that similar residuals do not necessarily mean that SDC and MRSDC give comparable accuracy, only that they approximate their respective collocation solution to roughly the same degree.

To demonstrate (ii), Figure 6 shows the measured order of convergence for a wide range of time steps for MRSDC with $(M, P) = (5, 2)$ nodes (dashed lines) and $(M, P) = (5, 8)$ nodes (solid lines). Color indicates the number of iterations, ranging from $k = 0$ (predictor only) up to $k = 4$ (predictor plus four correction sweeps). The exact solution against which we compare is computed by running the method with a time steps many orders of magnitude smaller. Since order is defined for step sizes approaching zero, we see some inconsistent behavior for larger time step sizes. With decreasing step size, however, measured order approaches the theoretically expected order of $\min(M, P, k + 1)$. This illustrates that MRSDC, just as single-rate SDC, improves formal order by one per iteration up to the order of the underlying quadrature rules.

5 Real-time simulation of the 3D fully coupled machine

We now demonstrate that MRSDC implemented in DUNE can accurately solve the 3D fully coupled problem accurately with look-ahead factors $\eta \gg 1$. Further, we show that the multi-rate time stepping produces a smoother temperature field and more accurate approximations of deformations. We focus on the error from the time discretization since the FEM approach used in space is standard and its analysis is now textbook material.

The physical parameters and geometries are the same as in Naumann et al. [27], except for the movement profile of the stock and the simulation time. We let the stock move according to

$$s(t) = a \sin\left(\frac{2\pi}{\varepsilon} t\right) + s_0, \tag{42}$$

in meters with a =0.495 m and s_0=0.505 m. The movement of the stock is periodic with each period having length $\varepsilon = 24$s. We simulate 10 periods for a final time of $T = 240$s. Our MRSDC time stepping uses $(M, P) = (3, 2)$ quadrature nodes and 24 time steps of length $\Delta t = 1.0$s per period. As initial data we use the stationary temperature profile for the machine at rest but subject to thermal coupling with the environment and cooling equipment

$$\mathbf{M} \vec{T}(0) = \vec{f}^{\,I}(\vec{T}(0)) + \vec{g}(0)\,. \tag{43}$$

Figure 7 shows the temperature field at the end of the simulation at $t = 4$min. The maximum temperature (up to 25.7 °C) is found at the center of rails. The

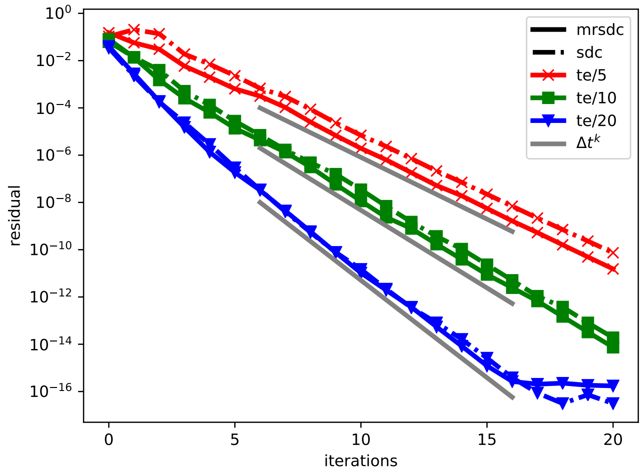


Fig. 5 Residual versus iteration count k for SDC with $M = 5$ nodes (dashed lines) and MRSDC with $(M, P) = (5, 8)$ nodes (solid lines). Both convergence to the collocation solution at approximately the same rate proportional to Δt^k (indicated by grey lines).

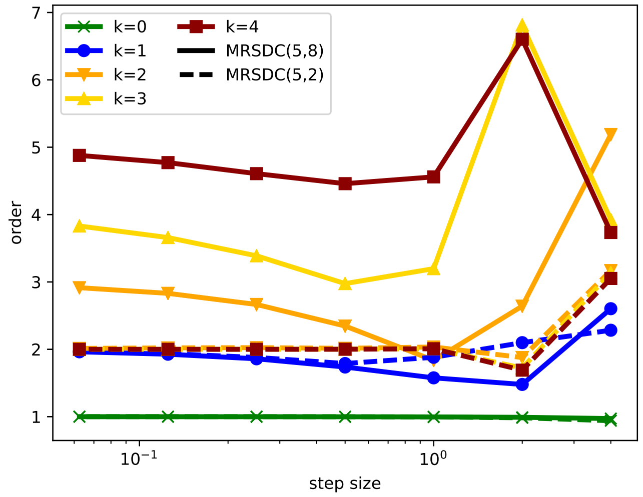


Fig. 6 Measured convergence order of MRSDC with $(M, P) = (5, 8)$ nodes (solid line) and $(M, P) = (5, 2)$ nodes (dashed lines) for $k = 0, 1, 2, 3, 4$ iterations. The order matches the theoretically expected value of $\min(M, P, k + 1)$. Note that for $(M, P) = (5, 2)$ computing more than $k = 2$ iterations does not increase the order further because $P = 2$ limits the order of the collocation method to two.

fact that we cool the right side of the stand with fixed temperature of 22.0 °C while the left side is exposed to room temperatures of up to 24.5 °C creates a slight asymmetry with the left rail being warmer. Minimum temperatures of 21.8 °C are found at the floor which has a temperature of 20 °C and thus removes heat from the machine. Temperatures at the top of the sides are somewhat higher than towards the bottom because of the small vertical temperature gradient in the environment and the cooling at the bottom.

The graphs on the right of Figure 7 illustrate the spatial and temporal variation of the temperature field in specific parts of the machine. The upper Figure shows the temperature along the indicated cross-section of the right rail at three different times. For reference, the temperature at $t = 8\frac{2}{3}\varepsilon$ computed with implicit Euler is shown as well. Toward the center of the rail (at around $y = 1.25$m), the heat generated with each passage of the stock slowly accumulates so that temperature is higher at later times. After $t = 9\frac{2}{3}$ periods, the temperature has increased by up to 1.4 °C above the reference temperature. The strongest warming is seen around the center and the temperature increase becomes less pronounced towards the ends of the rail. There, the longer time between passages of the stock leaves enough time for the heat to dissipate and only a small increase in temperature of about 0.1 °C is observed at the upper and lower end. Furthermore, at full periods, the stock is located at the center of the rail moving upwards, having just heated the lower part, while at two-third periods the stock is near the bottom moving downwards. This causes a slight shift in the temperature profile at times at full periods, i.e. t=9ε and $t = 10\varepsilon$, compared to times $t = \frac{2}{3}\varepsilon, \frac{4}{3}\varepsilon, \ldots$.

The lower Figure in 7 shows the transient effects from the moving stock. Each of the two transits per period (one while moving downwards, one while moving upwards) leads to an increase in temperature, followed by a more gradual decrease due to heat diffusion. Because the time without transits increases for points away from the center of the rail, there is a longer period of time for the heat to diffuse, leading to less warming. Transient profiles are therefore not the same throughout the machine but vary with spatial position and models that rely on a separation of spatial and temporal coordinates will not be able to capture this effect. Points near the center (2 and 3 in Figure 7) experience a significant net heating of 1.4 °C and 0.9 °C respectively over the course of the simulation. In contrast, points towards the ends of the rails (1 and 4 in Figure 7) only warm by about 0.1 °C to 0.2 °C. There is less heating toward the lower end of the rail because of the shorter distance to the cooling floor.

Deviations from the reference temperature create thermal deformation. Figure 8 shows the deformation of the stand resulting from the temperature field shown in Figure 7. Since deformations are of the order of 5 µm, they are exaggerated in the figure by a factor of 10^4 to make them visible. The stand mainly bends toward the rear and to the right with stronger deformations at the top. Because of the transient and inhomogeneous distribution of heat, deformations are not uniform but depend strongly on time and position. The upper right

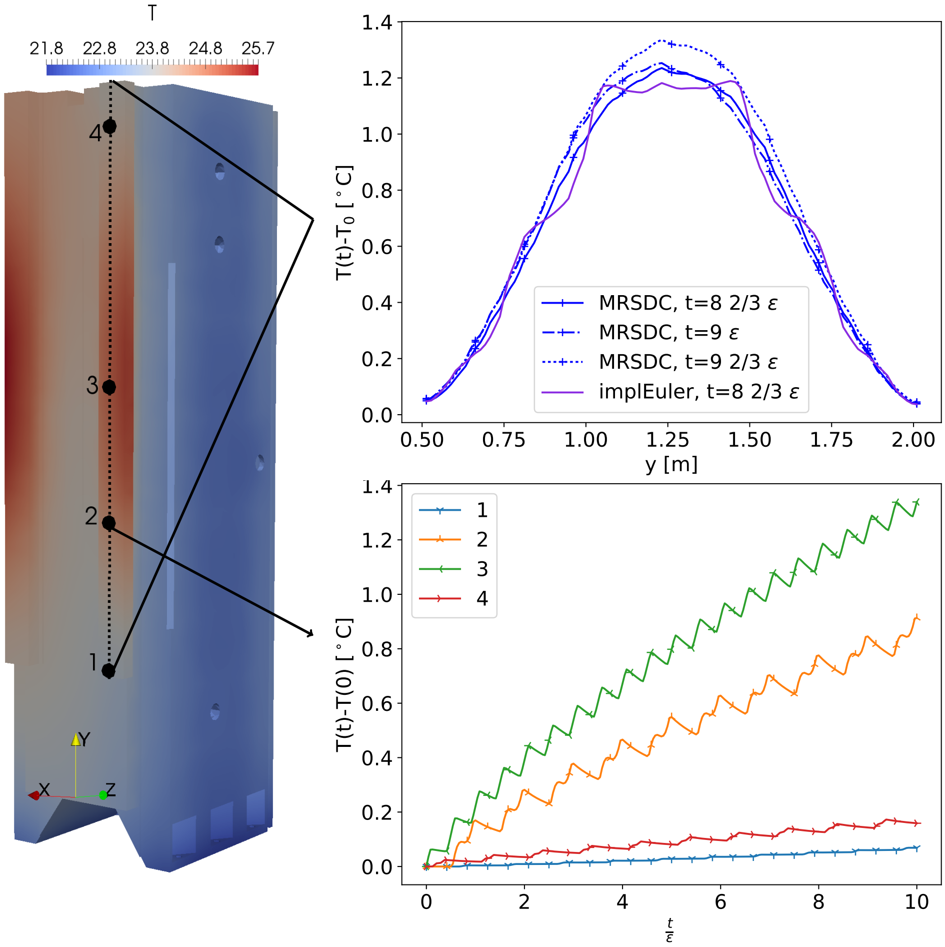


Fig. 7 Simulated temperature field on the fixed part of the machine after t =240 s simulated time (left figure). The upper right figure shows the cross-section of the temperature field along the black line on the rail at four different times. Note that at times $t = \frac{2}{3}\varepsilon, \frac{4}{3}\varepsilon, \ldots$, the stock is at the lower end of the rail. At times $t = \varepsilon, 2\varepsilon, \ldots$ it is at the centre of the rail. The lower right figure shows the temperature over time at the four points indicated by the black dot in the left figure.

Figure shows deformations along the cross-section of the right rail indicated in Figure 7. The vertical gradient of temperature from the bottom to the top creates significant deformation along the y-axis. Because most of the warming happens at the rails at the front, we observe substantial deformations in z direction. While relatively small toward the floor, both z and y deformation increase substantially towards the top. Deformations in x direction due to the slightly asymmetric warming are smaller with a maximum toward the center.

Next, we analyze performance of MRSDC in terms of work-precision. Figure 9 shows achieved time discretization errors (y-axis) in the temperature field versus the look-ahead factor (x-axis) which depends on the wall clock time required to run the simulation at this accuracy. Lower errors require better resolution which results in longer simulations and therefore smaller η. The threshold between faster than real-time ($\eta > 1$) and slower than real-time ($\eta < 1$) is indicated by a vertical black line. Three classes of method are investigated: multi-rate SDC (MRSDC, solid lines), single-rate SDC (dash-dotted lines) and implicit Euler (dashed line). MRSDC uses $(M, P) = (3, 2)$ nodes and $k = 0, 1, 2$ iterations while SDC uses $M = 3$ nodes and also up to two iterations. Note that the fast component $\mathbf{M}_{\mathbf{B}\text{fix}}(t)$ from (11) – which is treated explicitly with a small step in the multi-rate integrator – is included in the implicit part in single-rate SDC. This means that SDC requires the reassembly of the Jacobian that MRSDC avoids, creating substantial overhead.

For $k = 0$, both SDC and MRSDC show first order convergence in the faster and slower than real-time regime. Both methods deliver about the same efficiency, being slightly better than implicit Euler, but MRSDC has a slight advantage for large values of η. For higher order and $k = 1, 2$, single-rate SDC is substantially

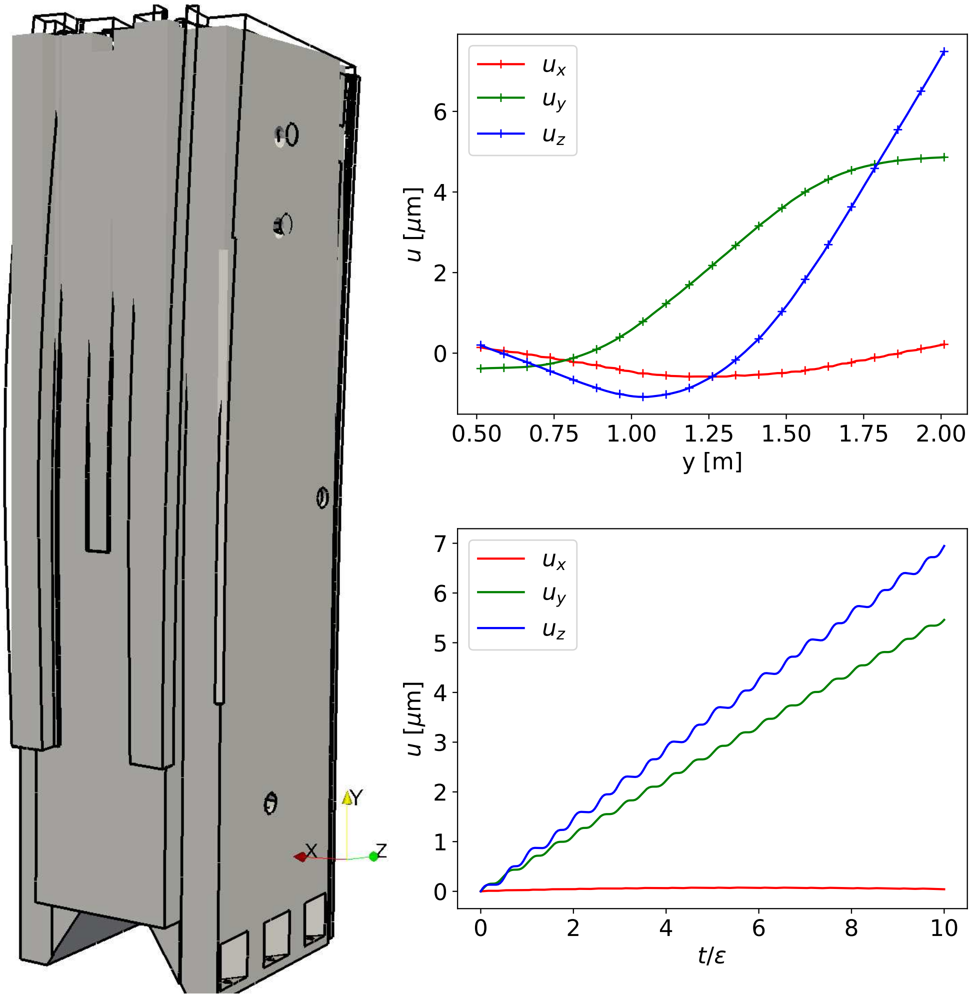


Fig. 8 The grey geometry shows the original stand whereas the lines show the deformed stand at time $t = 8\frac{2}{3}\varepsilon$ (deformations are exaggerated by a factor of $1e4$ for visibility). Heat is mainly generated at the rail, causing thermal expansion at the front, and therefore the stand bends predominantly in the z-direction, away from the rails. The top right figure shows deformations along the same line as in Figure 7 in x (red), y (green) and z (blue) direction. The bottom right figure shows the deformation in point 4 in Figure 7 over time.

less efficient than MRSDC, producing larger errors for the same η. Single-rate SDC is also mostly less efficient than implicit Euler except for values $\eta \ll 1$. MRSDC with $k = 1, 2$ is still in the pre-asymptotic regime for look-ahead factors larger than one, not yet showing the theoretical convergence order. A clear difference in the slopes of the error lines emerges only for factors of around $\eta \approx 1$ and smaller. Nevertheless, MRSDC with $k = 1, 2$ iterations is more efficient than first order MRSDC for most values between $\eta = 10$ to $\eta = 1$ and significantly more efficient than implicit Euler. Only for very coarse resolutions and values of $\eta > 10$ is there no clear gain from higher order MRSDC with all $k = 0, 1, 2$ showing roughly the same performance. Still, the multi-rate integration makes MRSDC more efficient than simple implicit Euler, delivering substantially more accurate solutions for the same look-ahead factors. For slower than real-time simulations with $\eta < 1$, MRSDC eventually shows its theoretical order of convergence which significantly widens the performance gap compared to first order implicit Euler.

We can relate the accuracy of the representation of the temperature field to the accuracy of deformations. Figure 10 shows the absolute error in the deformations in x, y and z-direction, computed from the gradients of the temperature field, along the cross-section indicated in Figure 7. Solid lines marked with circles correspond to MRSDC with $k = 1$ iteration and $\eta = 5.0$ while dashed lines indicate implicit Euler with $\eta = 6.5$. Clearly, the higher accuracy in the computed temperature field translate into significantly more accurate deformations, with errors from MRSDC being at least one order of magnitude smaller than those from implicit Euler.

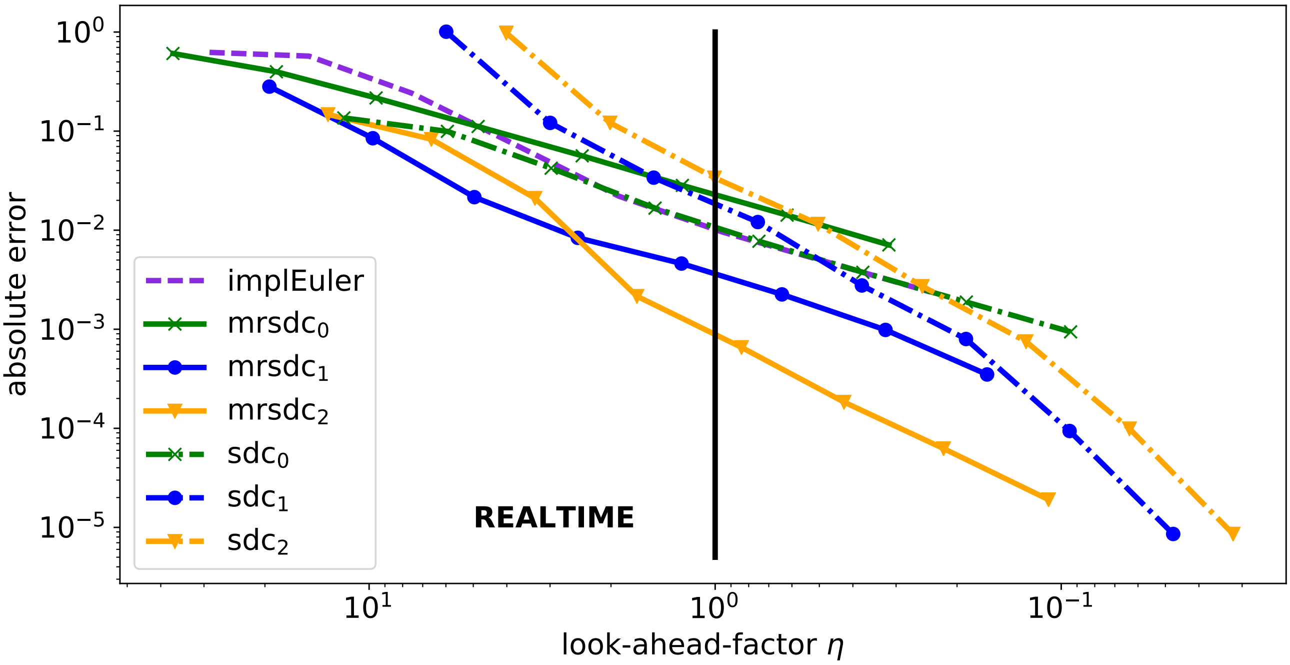


Fig. 9 Time-discretisation error for SDC (dash-dotted lines), MRSDC (straight lines) with M=3, P=2 at $t_{end} = 10\varepsilon$ and implicit euler (dashed line). The black vertical line indicates $\eta = 1.0$ with simulations on the left running with $\eta > 1$ or faster than real-time.

For the results shown above, we only computed deformations at the end of the simulation. In reality, one would have to compute the deformations more frequently. However, for linear elasticity, computing the deformations reduces to the solution of a large sparse linear system of equations with the temperature profile as right hand side. When computing the LU-decomposition of the coefficient matrix at the beginning of the simulation, solving for the deformations requires only a forward-backward solve of lower and upper triangular matrices. For the problem studied here, this took about 0.08 s which is negligible compared to the end time 240 s so that more frequent solves will have minimal effect on the reported look-ahead factors.

Stronger heating. It is worth pointing out that the heat generation in the studied problem is relatively mild but that our method is applicable to scenarios with stronger heating as well. We performed simulations with a 100× stronger heat source (not shown). Since the problem is linear, this essentially scales up all curves in Figure 10 by a factor of hundred, so that errors in the implicit Euler go up to 10 µm while errors from MRSDC go up to 0.1 µm. Because manufacturing tolerances in essence pose limits on the absolute error, stronger heating will therefore require tighter numerical tolerances which is likely going to favour higher order method like MRSDC. Analyzing a wider parameter range in more detail is left for future work.

6 Conclusions and outlook

The paper introduces a multi-rate high-order time stepping method for simulations of heat diffusion in moving machine tools consisting of a fixed stand and a moving stock. By implementing the algorithm in the open-source FEM framework DUNE, we demonstrate that accurate transient simulations of a FE model of the fully coupled machine are possible in real-time. We show that the higher order of multi-rate spectral deferred corrections (MRSDC) improves computational efficiency compared to implicit Euler, even for large time steps where the method does not yet achieve its theoretical order of accuracy. Time discretization errors of around one percent can be achieved for look-ahead factors of $\eta = 10$. The results illustrate the potential of solving FEM fast enough to deliver spatially and temporally resolved temperature fields for online compensation of errors due to thermal deformation.

Outlook. Open source libraries like the one used in this paper offer the possibility of significant further performance optimization. Making use of parallelization and high-powered accelerators like graphics processing units or many-core CPUs would require substantial effort but

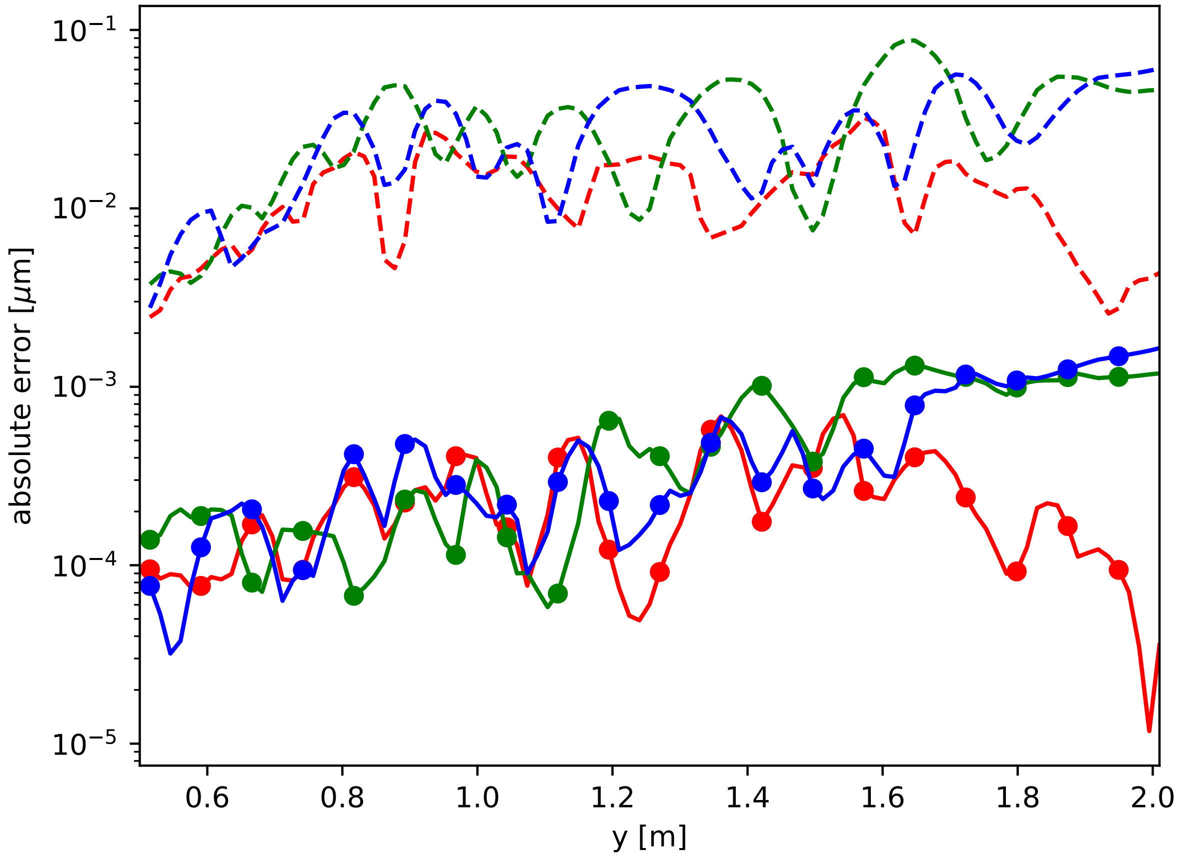


Fig. 10 Errors in the displacements in x (red), y (green) and z (blue) direction at time t=$8\frac{2}{3}\varepsilon$ along the line in figure 7. The displacement correspond to the temperatures in the right top image of Figure 7. Lines with markers show the MRSDC(3,2) solution with one iteration, dashed lines represent the implicit euler. The errors in the displacements from the MRSDC solutions are about two magnitudes smaller. Results are shown for simulations using 12 time steps per period, which corresponds to a look-ahead-factor $\eta \approx 5$ for both methods.

could likely increase look-ahead factor by another order of magnitude or more while maintaining high accuracy. Exploring novel strategies like parallelization in time [12] could increase η even further. This would eventually allow to use full FE model as part of filter-based approaches that combine model and measurements into best estimates of the state of a machine [33]. Furthermore, coupling a FE model with a suitable model for error compensation and validating it in a realistic experimental setting [31] would be an important next step.

Acknowledgments

This work was funded by the German Research Foundation as part of the CRC/TR 96. We thankfully acknowledge help from Ansgar Burchardt and Oliver Sander with the DUNE grid-glue library.

References

1. ANSYS® Academic Research. Release 15.
2. Appendix I - Matrix diagonalization or lumping. In O.C. Zienkiewicz, R.L. Taylor, and J.Z. Zhu, editors, *The Finite Element Method Set (Sixth Edition)*, pages 704 – 709. Butterworth-Heinemann, Oxford, sixth edition edition, 2005.
3. Ali M. Abdulshahed, Andrew P. Longstaff, and Simon Fletcher. The application of ANFIS prediction models for thermal error compensation on CNC machine tools. *Applied Soft Computing*, 27:158 – 168, 2015.
4. Peter Bastian, Gerrit Buse, and Oliver Sander. *Infrastructure for the Coupling of Dune Grids*, pages 107–114. Springer Berlin Heidelberg, Berlin, Heidelberg, 2010.
5. Anne Bourlioux, Anita T. Layton, and Michael L. Minion. High-order multi-implicit spectral deferred correction methods for problems of reactive flow. *Journal of Computational Physics*, 189(2):651 – 675, 2003.
6. Elizabeth L. Bouzarth and Michael L. Minion. A multirate time integrator for regularized Stokeslets. *Journal of Computational Physics*, 229(11):4208 – 4224, 2010.

7. Hongrui Cao, Xingwu Zhang, and Xuefeng Chen. The concept and progress of intelligent spindles: A review. *International Journal of Machine Tools and Manufacture*, 112:21 – 52, 2017.
8. Andrei Constantinescu and Alexander Korsunsky. *Elasticity with Mathematica : An Introduction to Continuum Mechanics and Linear Elasticity*. Cambridge University Press, 2007.
9. B Denkena and KH Scharschmidt. Kompensation thermischer verlagerungen. *Werkstatttechnik Online*, 97(11-12):913–917, 2007.
10. Zheng-Chun Du, Si-Yu Yao, and Jian-Guo Yang. Thermal behavior analysis and thermal error compensation for motorized spindle of machine tools. *International Journal of Precision Engineering and Manufacturing*, 16(7):1571–1581, 2015.
11. Alok Dutt, Leslie Greengard, and Vladimir Rokhlin. Spectral deferred correction methods for ordinary differential equations. *BIT Numerical Mathematics*, 40(2):241–266, 2000.
12. Matthew Emmett and Michael L. Minion. Toward an Efficient Parallel in Time Method for Partial Differential Equations. *Communications in Applied Mathematics and Computational Science*, 7:105–132, 2012.
13. Matthew Emmett, Weiqun Zhang, and John B. Bell. High-order algorithms for compressible reacting flow with complex chemistry. *Combustion Theory and Modelling*, 18:361 – 387, 2014.
14. Christian Engwer and Steffen Müthing. *Concepts for Flexible Parallel Multi-domain Simulations*, pages 187–195. Springer International Publishing, 2016.
15. Alexander Galant, Michael Beitelschmidt, and Knut Großmann. Fast high-resolution FE-based simulation of thermo-elastic behaviour of machine tool structures. *Procedia CIRP*, 46:627 – 630, 2016. 7th HPC 2016 CIRP Conference on High Performance Cutting.
16. Phillip L. Gould. *Formulation, Uniqueness, and Solution Strategies*, pages 91–100. Springer New York, New York, NY, 2013.
17. E. Hairer, S. P. Nørsett, and G. Wanner. *Solving Ordinary Differential Equations I: Nonstiff problems*. Springer-Verlag Berlin Heidelberg, 2nd edition, 1993.
18. Zhao Haitao, Yang Jianguo, and Shen Jinhua. Simulation of thermal behavior of a cnc machine tool spindle. *International Journal of Machine Tools and Manufacture*, 47(6):1003 – 1010, 2007.
19. Anita T. Layton and Michael L. Minion. Conservative multi-implicit spectral deferred correction methods for reacting gas dynamics. *Journal of Computational Physics*, 194(2):697 – 715, 2004.
20. J. W. Li, W. J. Zhang, G. S. Yang, S. D. Tu, and X. B. Chen. Thermal-error modeling for complex physical systems: the-state-of-arts review. *The International Journal of Advanced Manufacturing Technology*, 42(1):168, 2008.
21. Yang Li, Wanhua Zhao, Shuhuai Lan, Jun Ni, Wenwu Wu, and Bingheng Lu. A review on spindle thermal error compensation in machine tools. *International Journal of Machine Tools and Manufacture*, 95:20 – 38, 2015.
22. Yang Li, Wanhua Zhao, Wenwu Wu, and Bingheng Lu. Boundary conditions optimization of spindle thermal error analysis and thermal key points selection based on inverse heat conduction. *The International Journal of Advanced Manufacturing Technology*, 90(9):2803–2812, 2017.
23. Chi Ma, Liang Zhao, Xuesong Mei, Hu Shi, and Jun Yang. Thermal error compensation based on genetic algorithm and artificial neural network of the shaft in the high-speed spindle system. *Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture*, 231(5):753–767, 2017.
24. Josef Mayr. *Beurteilung und Kompensation des Temperaturganges von Werkzeugmaschinen*. PhD thesis, ETH Zürich, 2009.
25. Josef Mayr, Jerzy Jedrzejewski, Eckart Uhlmann, M. Alkan Donmez, Wolfgang Knapp, Frank Härtig, Klaus Wendt, Toshimichi Moriwaki, Paul Shore, Robert Schmitt, Christian Brecher, Timo Würz, and Konrad Wegener. Thermal issues in machine tools. *CIRP Annals - Manufacturing Technology*, 61(2):771 – 791, 2012.
26. Michael L. Minion. Semi-implicit spectral deferred correction methods for ordinary differential equations. *Communications in Mathematical Sciences*, 1(3):471–500, 2003.
27. Andreas Naumann, Norman Lang, Marian Partzsch, Michael Beitelschmidt, Peter Benner, Axel Voigt, and Jörg Wensch. Computation of thermo-elastic deformations on machine tools a study of numerical methods. *Production Engineering*, 10(3):253–263, 2016.
28. Marian Partzsch and Michael Beitelschmidt. An arithmetic correction for avoiding non-uniform heat input distribution caused by translational motions within time-discrete thermal analyses. *Volume 3: Engineering Systems; Heat Transfer and Thermal Engineering; Materials and Tribology; Mechatronics; Robotics*, 2014.
29. Daniel Ruprecht and Robert Speck. Spectral deferred corrections with fast-wave slow-wave splitting. *SIAM Journal on Scientific Computing*, 38(4):A2535–A2557, 2016.
30. Oliver Sander. Getting started with DUNE. 2017.
31. Xaver Thiem, Mirko Riedel, Bernd Kauschinger, and Jens Müller. Principle and verification of a structure model based correction approach. *Procedia CIRP*, 46:111 – 114, 2016. 7th HPC 2016 CIRP Conference on High Performance Cutting.
32. Martin Weiser. Faster SDC convergence on non-equidistant grids by DIRK sweeps. *BIT Numerical Mathematics*, 55(4):1219–1241, 2014.
33. Simon Zuest, Philip Paul, Lukas Weiss, and Konrad Wegener. Modelling and compensation of thermally induced positioning errors in a high precision positioning application. *IFAC-PapersOnLine*, 49(21):347 – 353, 2016. 7th IFAC Symposium on Mechatronic Systems MECHATRONICS 2016.