

FIRST LOOK AT SPACETIME

Let me keep it short. We did a first run of the spacetime project solving the heat equation on the unit square $\Omega = [0, 1]^2$ on the time interval $I := [0, 1]$. The true solution is

$$u(t, x, y) := (t^2 + 1)(1 - x)x(1 - y)y.$$

Given some level L , we build X_δ as the double tree consisting of all double nodes $\lambda = (\lambda_{\text{time}}, \lambda_{\text{space}})$ for which $2|\lambda_{\text{time}}| + |\lambda_{\text{space}}| \leq L$ (a *sparse grid*). We then build Y_δ through the formula halfway page 6 of [followup.pdf](#). We use the system on the bottom of page 6, and compute the data integrals in the right-hand side using Gauss quadrature of sufficient order to be exact.

Denote the 2×2 block-matrix with M . We use MINRES with initial guess x_0 equal to the right-hand side b of the system, and continue until the relative residual is ‘small’:

$$\frac{\|Mx_k - b\|_2}{\|x_k\|_2} \leq 10^{-5}.$$

This gives us a discrete solution (μ_δ, u_δ) for which we can measure

$$E(t) := t \mapsto \|\gamma_t(u_\delta - u)\|_{L_2(\Omega)}.$$

We will use this as a proxy for the error $\|u - u_\delta\|_X$ which is difficult to measure.

Given X_δ , we distinguish its dimension $\dim X_\delta$ and $\#X_\delta$, being the total number of *degrees of freedom*, i.e., the dimension of X_δ minus the number of Dirichlet double nodes (double nodes for which the vertex of λ_{space} is not on the domain boundary).

CONVERGENCE RATE

The code right now seems to solve the problem, as can be seen by Figure 1: Increasing L (thereby increasing $\#X_\delta$, yields an error reduction. We discern a pre-asymptotic regime: the error hardly decreases between levels $L = 1$ and 2. Moreover, errors seem to taper off near the final levels L , which may have something to do with our solver tolerance.

In fact, the errors decay at an algebraic rate somewhere between 0.75 and 0.85, depending on the time t at which the errors are measured, as well as whether or not we keep the full range of errors when calculating the rate (for instance, in the brown line, we slice away level 1 because it is in the pre-asymptotic regime, and levels 11 and 12 because the convergence stagnates in this regime).

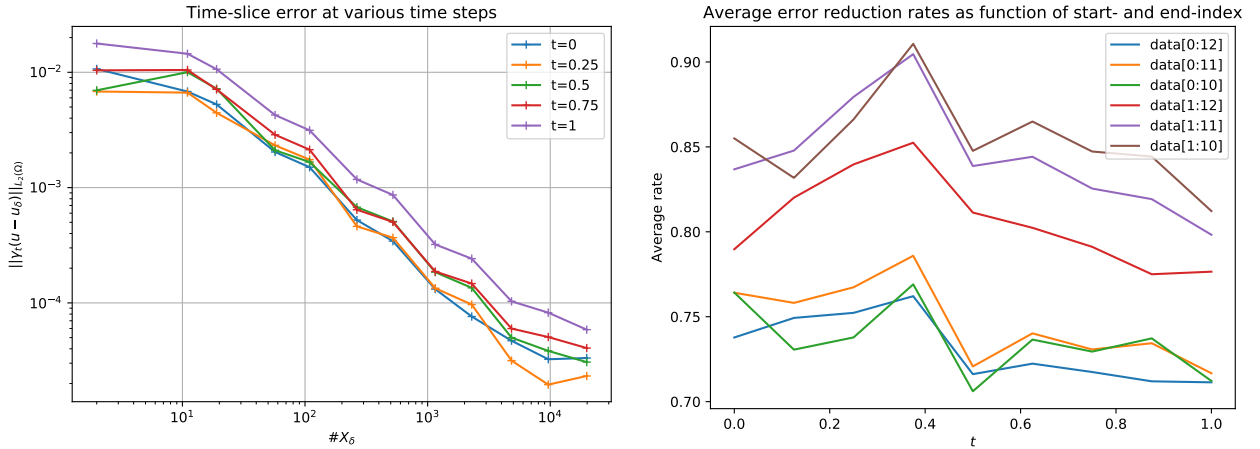


FIGURE 1. Left: Time slice errors. Right: rates.

COMPUTATIONAL COST

Let's measure the time it takes to perform a single application of the system matrix. We do not take into account the setup cost (like computing Σ and Θ). Figure 2 paints a slightly demotivating picture, in that the bottom graph is not constant. We do have to place an asterisk here: for small systems, the produced numbers are averaged over a small number of iterations, so they change a lot from run to run.

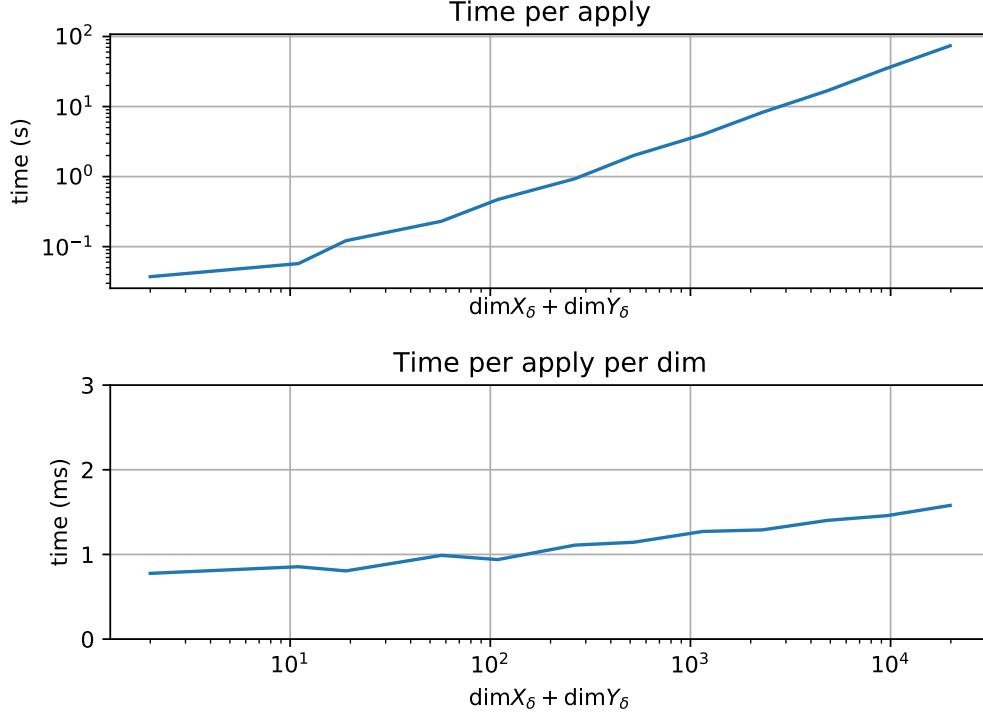


FIGURE 2. Top: time (s) per apply; bottom: time (ms) per apply per DoF.

THE SOLVER

As mentioned before, we use MINRES with default parameters (tolerance 10^{-5} , initial guess x_0 equal to the RHS b). In Figure 3, we show the residual norm $\|Mx_k - b\|_2$ as a function of k for a selection of solves. We see that the number of iterations increases strongly, suggesting that we could definitely use a good preconditioner.

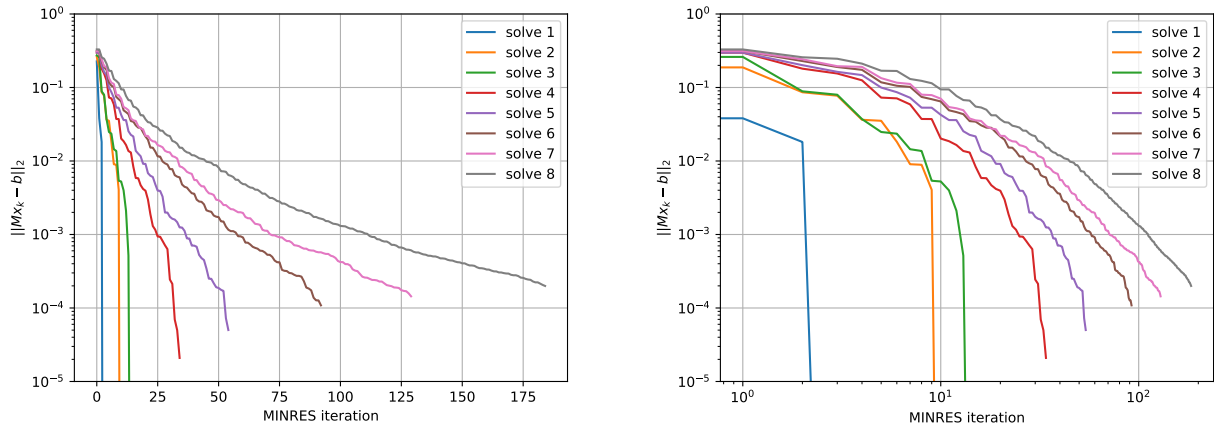


FIGURE 3. Left: semi-log-y MINRES convergence graph for some solves. Right: Same results in log-log format.