

## 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  $X_\delta$  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  $Y_\delta$  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  $2 \times 2$  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 norm is ‘small’:

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

This gives us a discrete solution  $(\mu_\delta, u_\delta)$  for which we can measure

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

by computing

$$E(t)^2 = \langle u(t), u(t) \rangle_{L_2(\Omega)} + \langle u_\delta(t), u_\delta(t) \rangle_{L_2(\Omega)} - 2\langle u(t), u_\delta(t) \rangle_{L_2(\Omega)}$$

where we have the first term analytically, the second number can be computed by an applicator of the space-time mass matrix, and the final term is found using quadrature of sufficient degree. We will use this as a proxy for the error  $\|u - u_\delta\|_X$ , which is difficult to measure.

Given  $X_\delta$ , we distinguish the dimension  $\dim X_\delta$  of the function space and  $\#X_\delta$ , being the total number of double nodes in the double tree.

### CONVERGENCE RATE

The code right now seems to solve the problem, as can be seen by Figure 1: Increasing  $L$  (thereby increasing  $\dim X_\delta$ , yields an error reduction. We discern a pre-asymptotic regime: the error hardly decreases between levels  $L = 1$  and 2. Moreover, errors seem to stagnate/ 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).

### 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  $\Sigma$  and  $\Theta$ ). Figure 2 reveals a modest increase in the time spent per apply per doubler node, which hints at a very slight superlinearity in the computational cost of a single apply. However, we see only a factor-2 increase over the course of the simulation.

### 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 *absolute* 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.

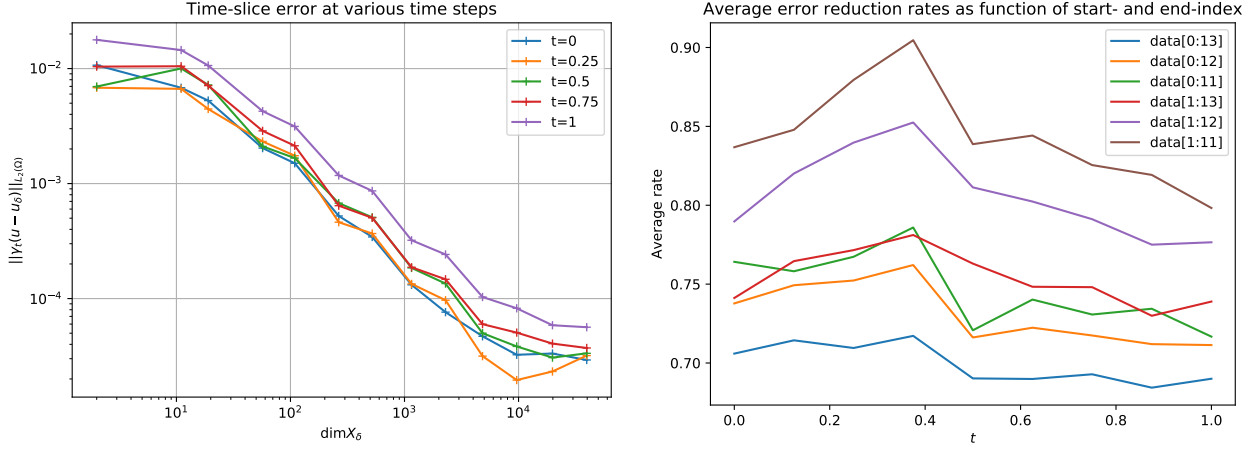


FIGURE 1. Left: Time slice errors. Right: rates.

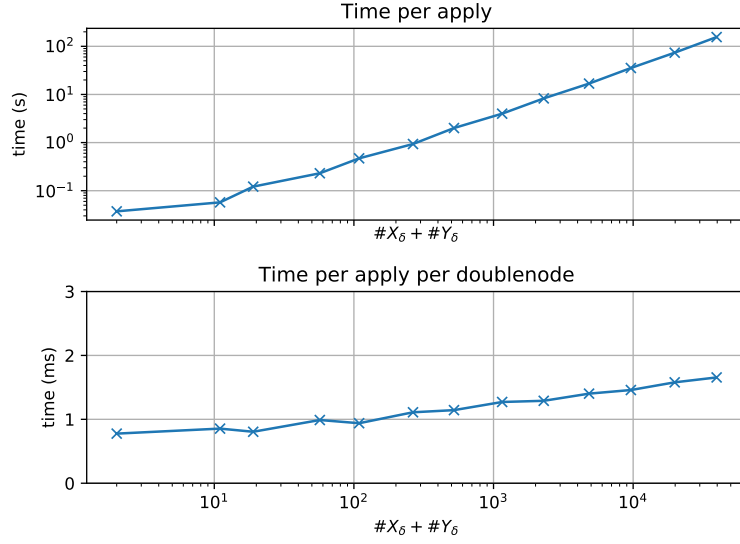


FIGURE 2. Top: time (s) per apply; bottom: time (ms) per apply per DoF.

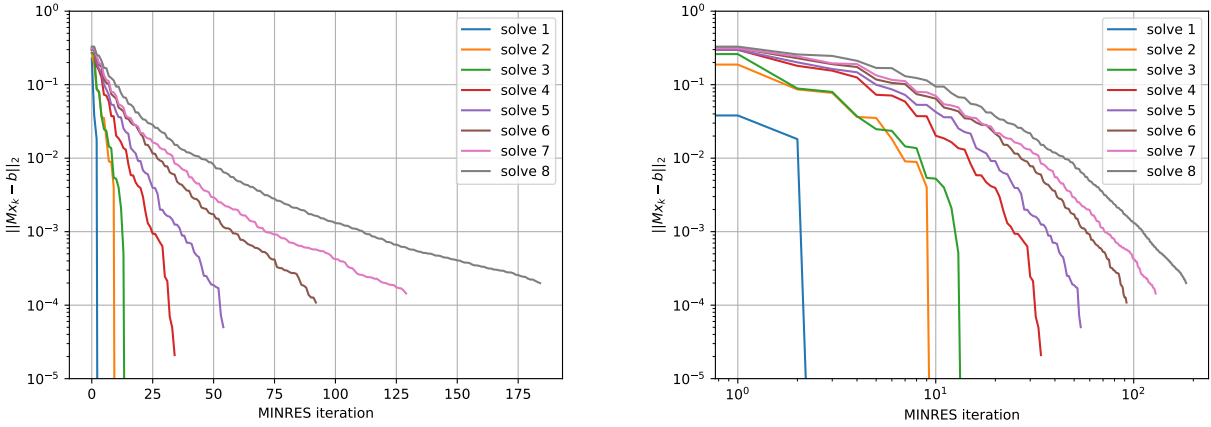


FIGURE 3. Left: semi-log-y MINRES convergence graph for some solves. Right: Same results in log-log format.