AM225: Assignment 3 solutions[1]

1. **Strassen's algorithm.** In Table 1, we show timing results for each method. In Figure 1, we plot those results along with fit parameters to a distribution $T = aN^b$, where two matrices of size $N \times N$ require time T to multiply. As expected, BLAS is fastest over a wide range of matrix sizes, but has worse scaling than Strassen's algorithm.

N	Strassen	standard	BLAS
4	6.04×10^{-7}	6.11×10^{-8}	7.35×10^{-8}
8	4.44×10^{-6}	3.10×10^{-7}	3.07×10^{-7}
16	3.25×10^{-5}	2.32×10^{-6}	1.99×10^{-6}
32	2.31×10^{-4}	2.36×10^{-5}	1.95×10^{-5}
64	8.61×10^{-4}	1.96×10^{-4}	1.31×10^{-4}
128	6.10×10^{-3}	1.61×10^{-3}	9.50×10^{-4}
256	4.30×10^{-2}	3.01×10^{-2}	7.48×10^{-3}
512	0.302	0.298	5.91×10^{-2}
1024	2.12	2.28	0.470
2048	15.0	75.6	5.40
4096	104	650	46.0
8192	732	5757	370

Table 1: Timing results, in seconds, for the three multiplication methods. As expected, BLAS is the fastest method accross a large number of possible matrix sizes

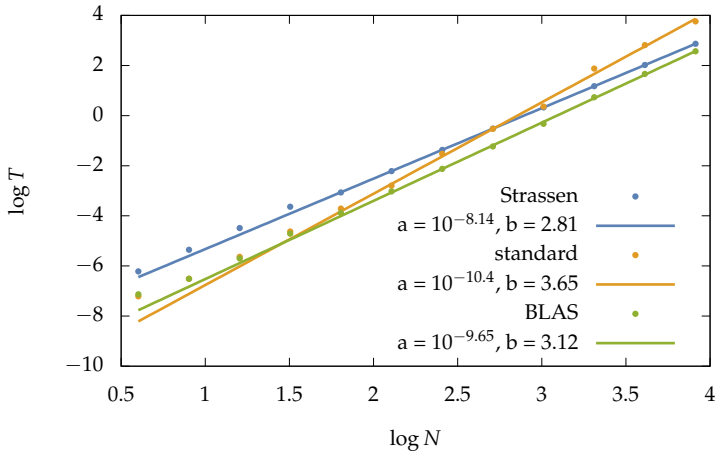


Figure 1: Timing results for the three multiplication methods, such that two matrices of size $N \times N$ take time T to multiply, fit to a distribution $T = aN^b$.

[1]Solutions to problems 1 and 2 by Nick Derr. Solutions to problems 3-5 written by Dan Fortunato.

2. **Fractal-based preconditioning.** Figure 2 shows the beneficial effects of choosing a point-sorting method to ensure dense Jacobi blocks. The top part of the figure shows that using the Hilbert curve does in fact produce denser blocks, and the bottom part shows that the resulting solve using PCG takes less time to converge.

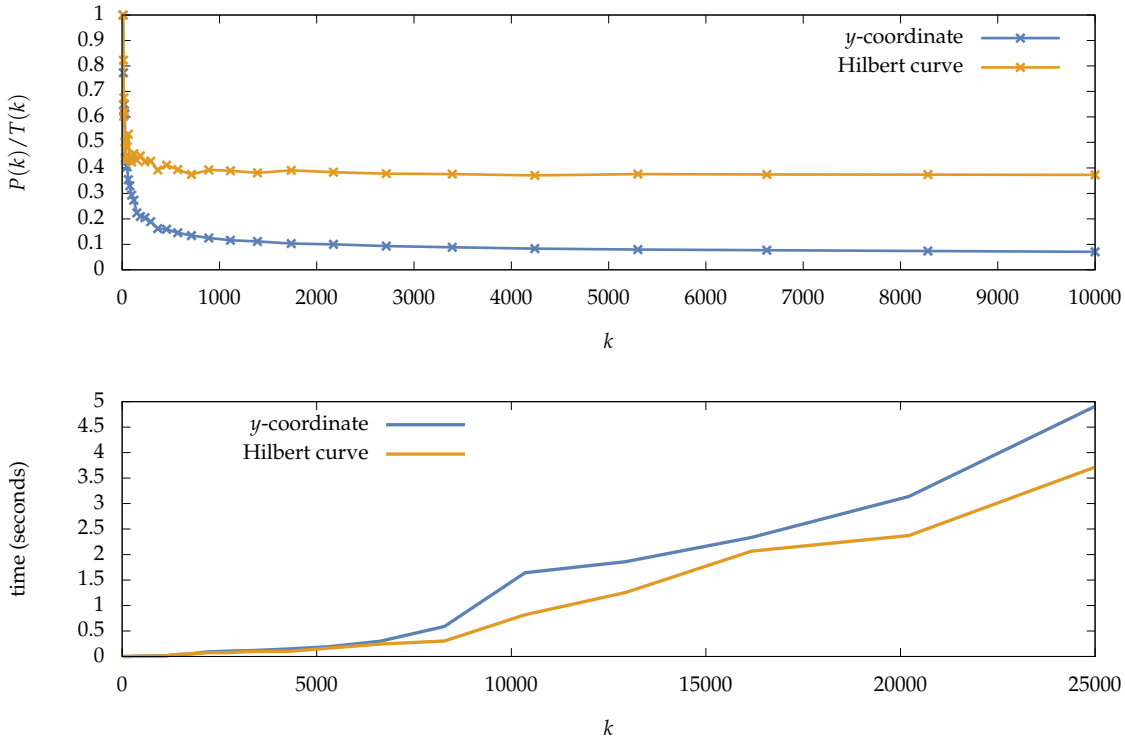


Figure 2: (Top) The fraction of nonzero entries in Jacobi blocks to total nonzero matrix entries is plotted against the grid size k for two methods of sorting points: 1) in order of rising y-coordinate and 2) position along the Hilbert curve. (Bottom) The time required for solution using the preconditioned conjugate gradient method is plotted for k in both cases.

3. **Alternative cubic elements** We consider solving the elliptic problem

$$\frac{d}{dx}\left(x\frac{du}{dx}\right) = f(x) \tag{1}$$

for a function $u(x)$ on $\Omega = (1,2)$ with the boundary conditions

$$u(1) = 0, \qquad \left.\frac{du}{dx}\right|_{x=2} = g \tag{2}$$

where g is a real constant.

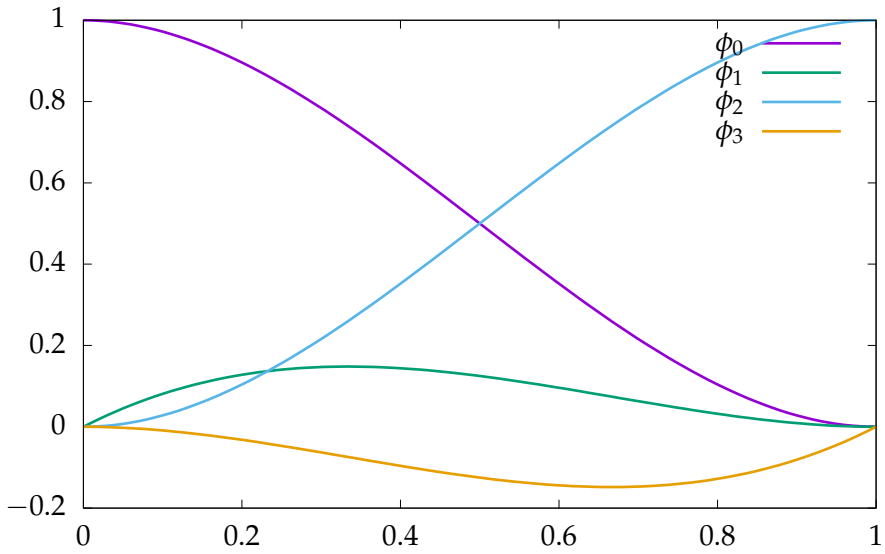


Figure 3: The local basis functions are cubic Hermite splines.

(a) The local finite element basis functions ϕ on the interval $[0, 1]$ satisfy

$$\begin{aligned}
&\phi_0(0) = 1, \quad \phi_0'(0) = 0, \quad \phi_0(1) = 0, \quad \phi_0'(1) = 0 \\
&\phi_1(0) = 0, \quad \phi_1'(0) = 1, \quad \phi_1(1) = 0, \quad \phi_1'(1) = 0 \\
&\phi_2(0) = 0, \quad \phi_2'(0) = 0, \quad \phi_2(1) = 1, \quad \phi_2'(1) = 0 \\
&\phi_3(0) = 0, \quad \phi_3'(0) = 0, \quad \phi_3(1) = 0, \quad \phi_3'(1) = 1
\end{aligned}$$

That is, the functions are analogues of the nodal basis functions in both value and derivative space. Such functions are cubic Hermite splines, given by

$$\begin{aligned}
\phi_0(x) &= 2x^3 - 3x^2 + 1, \\
\phi_1(x) &= x^3 - 2x^2 + x, \\
\phi_2(x) &= -2x^3 + 3x^2, \\
\phi_3(x) &= x^3 - x^2.
\end{aligned}$$

On neighboring elements the local basis functions ϕ can be matched together at the endpoints of each interval, where both functions have either value 1 or derivative 1. This gives the global finite element basis, ψ.

A plot of the finite element basis functions for $N = 3$ is shown in Fig. 4. When N intervals are used, the dimension of the finite element function space is $2N + 1$ (the additional function ψ_0 is not considered since it is constrained to zero by the Dirichlet condition at $x = 1$).

(b) Following the discussion in the lectures, a classical solution of Eqs. 1 & 2 will satisfy the variational problem

$$J(v) = \frac{1}{2}a(v, v) - (f, v)_{0,\Omega} - (g, v)_{0,\Gamma} \to \min \tag{3}$$

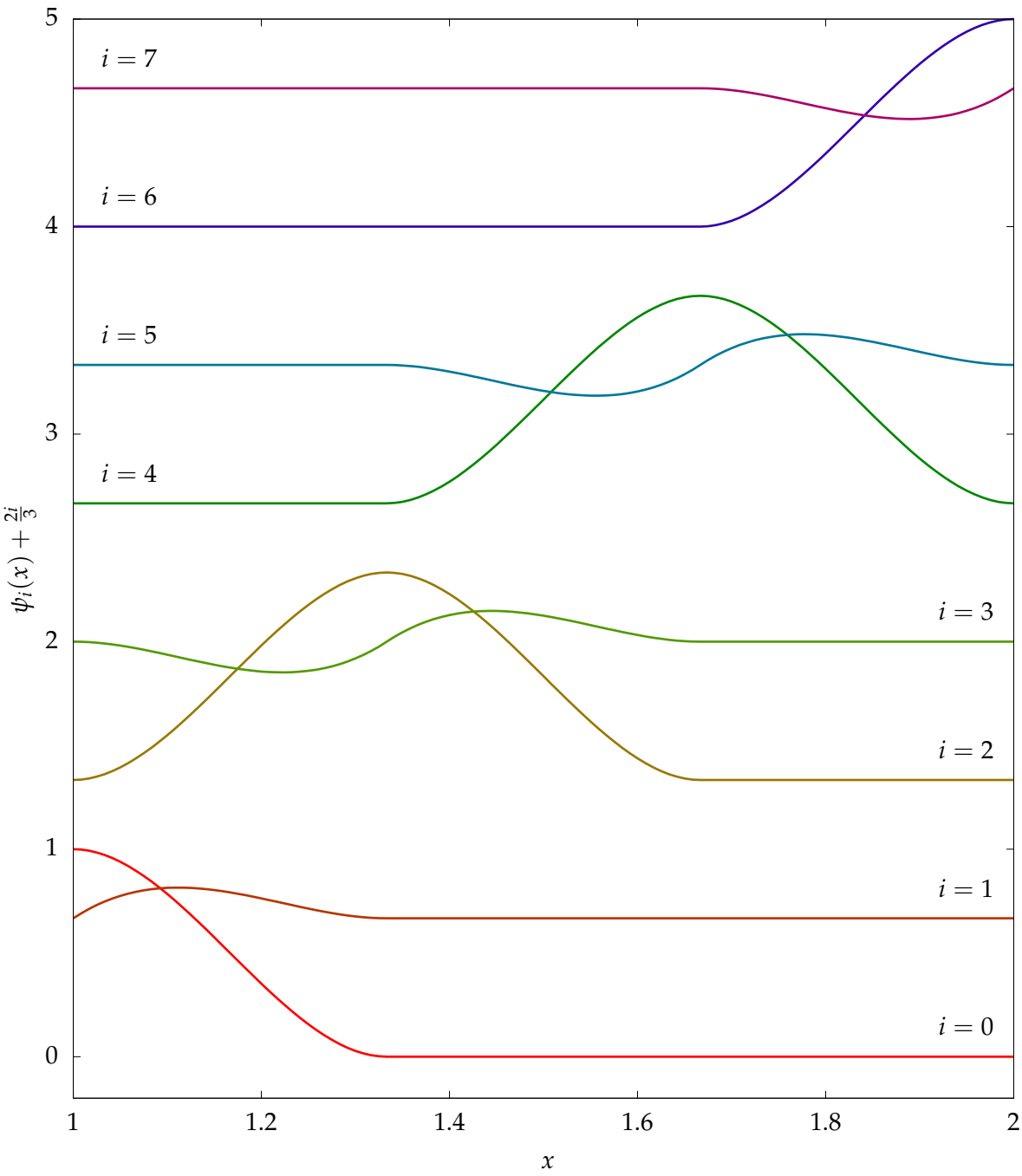


Figure 4: An example of the alternative cubic basis considered in question 1 that consists of C^1 piecewise cubic elements. The basis is illustrated for the case of three intervals over the range $[1, 2]$.

where

$$a(u,v) = \int_1^2 xu'v'\,dx, \qquad (f,v)_{0,\Omega} = \int_1^2 fv\,dx, \tag{4}$$

and the minimization is taken over all functions that satisfy the essential (Dirichlet) boundary condition that $v(1) = 0$. Here, the boundary Γ where natural (Neumann) boundary conditions are imposed consists of the single point at $x = 2$. Hence

$$(g,v)_{0,\Gamma} = gv(2). \tag{5}$$

We split the domain into N intervals $I_q = [x_q, x_{q+1}]$ for $q = 0,\ldots,N-1$. Using the Ritz–Galerkin method, the solution is written as

$$u_h(x) = \sum_{k=1}^{2N+1} z_k \psi_k(x) \tag{6}$$

and must satisfy

$$a(u_h,\psi_i) = \langle l,\psi_i\rangle \tag{7}$$

for all $i = 1,\ldots,2N+1$. Here $\langle l,\psi_i\rangle = (f,\psi_i)_{0,\Omega} + (g,\psi_i)_{0,\Gamma}$. To solve this, we first evaluate the terms $a(\psi_k,\psi_i)$ that form the stiffness matrix. Consider two distinct basis functions that overlap in an interval I_q. Let $k = q+\alpha$ and $i = q+\beta$ for $\alpha,\beta \in \{0,1,2,3\}$. Then in the interval I_q,

$$\psi_k(x) = \phi_\alpha\left(\frac{x-1}{h} - q\right), \qquad \psi_i(x) = \phi_\beta\left(\frac{x-1}{h} - q\right). \tag{8}$$

The derivatives of the cubic Hermite splines are

$$\begin{aligned}
\phi_0'(x) &= 6x^2 - 6x,\\
\phi_1'(x) &= 3x^2 - 4x + 1,\\
\phi_2'(x) &= 6x - 6x^2,\\
\phi_3'(x) &= 3x^2 - 2x.
\end{aligned}$$

Hence, using the substitution $x = (q+z)h + 1$,

$$\begin{aligned}
a(\psi_k,\psi_i) &= \int_{I_q} x\psi_k'\psi_i'dx = \frac{1}{h}\int_0^1 (qh + zh + 1)\phi_\alpha'(z)\phi_\beta'(z)dz\\
&= (q + h^{-1})B_{\alpha\beta} + C_{\alpha\beta}
\end{aligned} \tag{9}$$

where

$$B_{\alpha\beta} = \int_0^1 \phi_\alpha'(z)\phi_\beta'(z)dz, \qquad C_{\alpha\beta} = \int_0^1 z\phi_\alpha'(z)\phi_\beta'(z)dz. \tag{10}$$

The terms $B_{\alpha\beta}$ and $C_{\alpha\beta}$ are elementary integrals that can be performed using Mathematica. Their values are shown in Table 2.

Let us assume that the function f has the expansion

$$f(x) = \sum_{k=0}^{2N+1} f_k \psi_k(x). \tag{11}$$

$B_{\alpha\beta}$	$\alpha=0$	$\alpha=1$	$\alpha=2$	$\alpha=3$
$\beta=0$	$\frac{6}{5}$	$\frac{1}{10}$	$-\frac{6}{5}$	$\frac{1}{10}$
$\beta=1$	$\frac{1}{10}$	$\frac{2}{15}$	$-\frac{1}{10}$	$-\frac{1}{30}$
$\beta=2$	$-\frac{6}{5}$	$-\frac{1}{10}$	$\frac{6}{5}$	$-\frac{1}{10}$
$\beta=3$	$\frac{1}{10}$	$-\frac{1}{30}$	$-\frac{1}{10}$	$\frac{2}{15}$

$C_{\alpha\beta}$	$\alpha=0$	$\alpha=1$	$\alpha=2$	$\alpha=3$
$\beta=0$	$\frac{3}{5}$	$\frac{1}{10}$	$-\frac{3}{5}$	0
$\beta=1$	$\frac{1}{10}$	$\frac{1}{30}$	$-\frac{1}{10}$	$-\frac{1}{60}$
$\beta=2$	$-\frac{3}{5}$	$-\frac{1}{10}$	$\frac{3}{5}$	0
$\beta=3$	0	$-\frac{1}{60}$	0	$\frac{1}{10}$

Table 2: The terms $B_{\alpha\beta}$ (left) and $C_{\alpha\beta}$ that are used to assemble the stiffness matrix in the finite element calculation.

	$\alpha=0$	$\alpha=1$	$\alpha=2$	$\alpha=3$
$\beta=0$	$\frac{13}{35}$	$\frac{11}{210}$	$\frac{9}{70}$	$-\frac{13}{420}$
$\beta=1$	$\frac{11}{210}$	$\frac{1}{105}$	$\frac{13}{420}$	$-\frac{1}{140}$
$\beta=2$	$\frac{9}{70}$	$\frac{13}{420}$	$\frac{13}{35}$	$-\frac{11}{210}$
$\beta=3$	$-\frac{13}{420}$	$-\frac{1}{140}$	$-\frac{11}{210}$	$\frac{1}{105}$

Table 3: The terms $D_{\alpha\beta}$ that are used to assemble the source term in the finite element calculation.

Then to evaluate $(f,\psi_i)_{0,\Omega}$ we must evaluate $(\psi_k,\psi_i)_{0,\Omega}$. Again, choose distinct k and i so that their basis function overlap in an interval I_q and define $k=q+\alpha$ and $i=q+\beta$ for $\alpha,\beta\in\{0,1,2,3\}$. Then

$$(\psi_k,\psi_i)_{0,\Omega}=\int_{I_q}\psi_k\psi_i dx=h\int_0^1\phi_\alpha(z)\phi_\beta(z)dz=hD_{\alpha\beta} \tag{12}$$

where

$$D_{\alpha\beta}=\int_0^1\phi_\alpha(z)\phi_\beta(z)dz. \tag{13}$$

The values of $D_{\alpha\beta}$ are shown in Table 3.

With these calculations in place, we can formulate a linear system

$$Au=b \tag{14}$$

where Eq. 9 is used to assemble A, and Eq. 12 is used to assemble b. In addition, for the line corresponding to ψ_{2N+1} a contribution from Eq. 5 is included for $(g,v)_{0,\Gamma}$.

To test the convergence of the method, we use the method of manufactured solutions and propose that

$$u(x)=e^{1-x}\sin 5\pi x. \tag{15}$$

Then

$$u'(x)=e^{1-x}\left(5\pi\cos 5\pi x-\sin 5\pi x\right) \tag{16}$$

and therefore $g=u'(2)=e^{-1}5\pi$. In addition

$$f(x)=\frac{d}{dx}\left(xu'(x)\right)=e^{1-x}\left(5\pi(1-2x)\cos 5\pi x+\left((1-25\pi^2)x-1\right)\right)\sin 5\pi x\Big). \tag{17}$$

A log–log plot of the L_2 error as a function of N for a range of values from $N=10$ to $N=1000$ is shown in Fig. 5. Also shown is the L_2 error for the original cubic nodal basis.

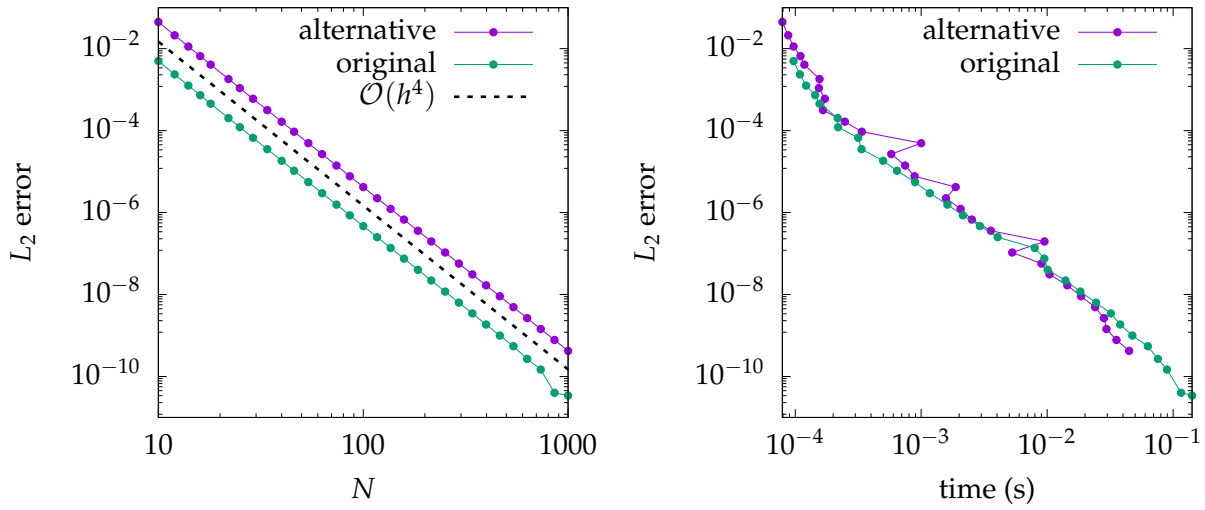


Figure 5: L_2 error versus N (left) and wall clock time (right) for both the alternative cubic basis and the original cubic basis.

(c) A log–log plot of the L_2 error for the two methods versus the wall clock time for the computation is shown in Fig. 5. The bases seem to be equally efficient in achieving a desired level of accuracy.

4. **Simulating an MRI** We model an MRI with the Helmholtz equation

$$\nabla^2 u(\mathbf{x}) + k^2 u(\mathbf{x}) = v(\mathbf{x}), \tag{18}$$

where $u(\mathbf{x})$ is the electric field strength, k is the wave number, and $v(\mathbf{x})$ is the electric excitation. In general, both $u(\mathbf{x})$ and k can be complex-valued. Note that $k^2 = \omega^2 \mu \epsilon$, where $\omega = 2\pi f$, μ is the magnetic permeability, and ϵ is the electric permittivity. In free space, $k^2 = k_0^2 = \omega^2 \mu_0 \epsilon_0$, where $\mu_0 = 4\pi \times 10^{-7}\,\mathrm{H\,m^{-1}}$ and $\epsilon_0 = 8.8542 \times 10^{-12}\,\mathrm{F\,m^{-1}}$.

(a) We use a second-order accurate finite difference scheme for Eq. 18 in free space on the domain $[0,1]^2$ with homogeneous Dirichlet boundary conditions. We set $v(\mathbf{x})$ to be the impulse $v(\mathbf{x}) = \delta(\mathbf{x} - (0.5, 0.5))$, discretized to a vector of zeros with a single element of $1/h^2$ at the appropriate row. We use a frequency of $f = 21.3\,\mathrm{MHz}$ and assume $u(\mathbf{x})$ is complex-valued.

To solve this problem with the banded solver, we write the discretized equations as

$$A\mathbf{u} = \mathbf{v}$$

where $\mathbf{u}$ and $\mathbf{v}$ are flattened versions of $u(\mathbf{x})$ and $v(\mathbf{x})$ sampled on the finite difference

grid,

$$A = \frac{1}{h^2}(K \otimes I + I \otimes K) + k_0^2(I \otimes I), \qquad K = \begin{bmatrix} -2 & 1 & & & \\ 1 & -2 & 1 & & \\ & \ddots & \ddots & \ddots & \\ & & 1 & -2 & 1 \\ & & & 1 & -2 \end{bmatrix},$$

and K and I are $(N-1) \times (N-1)$.

To solve this problem with the DST, we write the discretized equations as

$$\frac{1}{h^2}(XK + KX) + k_0^2 X = F$$

where X and F are $(N-1) \times (N-1)$ matrices containing the values of $u(\mathbf{x})$ and $v(\mathbf{x})$ sampled on the finite difference grid. We can share the k_0^2 component between the first two matrices to make this amenable to the DST:

$$\frac{1}{h^2}\left[X\left(K + \frac{k_0^2}{2}I\right) + \left(K + \frac{k_0^2}{2}I\right)X\right] = F.$$

Now we can perform an eigendecomposition on $K + \frac{k_0^2}{2}I$:

$$K + \frac{k_0^2}{2}I = S^{-1}\left(\Lambda + \frac{k_0^2}{2}I\right)S,$$

where S is the DST matrix and Λ is a diagonal matrix whose entries are the eigenvalues of K, given by $\lambda_j = -2 + 2\cos(j\pi/N)$. After some manipulation, this leads to

$$\frac{1}{h^2}\left[\left(K + \frac{k_0^2}{2}I\right)Y + Y\left(K + \frac{k_0^2}{2}I\right)\right] = SFS^{-1}$$

where $Y = SXS^{-1}$. Note that the multiplications by S and S^{-1} can be computed fast using the DST.[2]

The timings for the banded and DST-based solvers in free space are shown in Figure 6. Note that the DST-based solver is not faster than the banded solver until about $N > 64$.

(b) The MRIs of the human head using the banded solver with $N = 256$ and $v(\mathbf{x}) = \delta(\mathbf{x} - (0.6, 0.7))$ are shown in Figures 7a and 7b for frequencies of $f = 21.3\,\text{MHz}$ and $f = 298.3\,\text{MHz}$, respectively. The plots show the values of $|u|$. The solution has been clipped to the head for visualization purposes.

(c) The DST-based method no longer works if $k(\mathbf{x})$ is spatially-varying. This is because the eigenvectors of the operator $\nabla^2 + k(\mathbf{x})^2$ are no longer given by a sine series; in fact, we don't know what the eigenvectors are! Therefore, the operator is not diagonalizable by the DST. In the case of constant k, the operator $\nabla^2 + k^2$ has the same eigenvectors as ∇^2, but the eigenvalues are shifted by $k^2/2$.

[2] Note that for complex numbers $a + bi$, $\text{DST}(a + bi) = \text{DST}(a) + \text{DST}(b)i$.

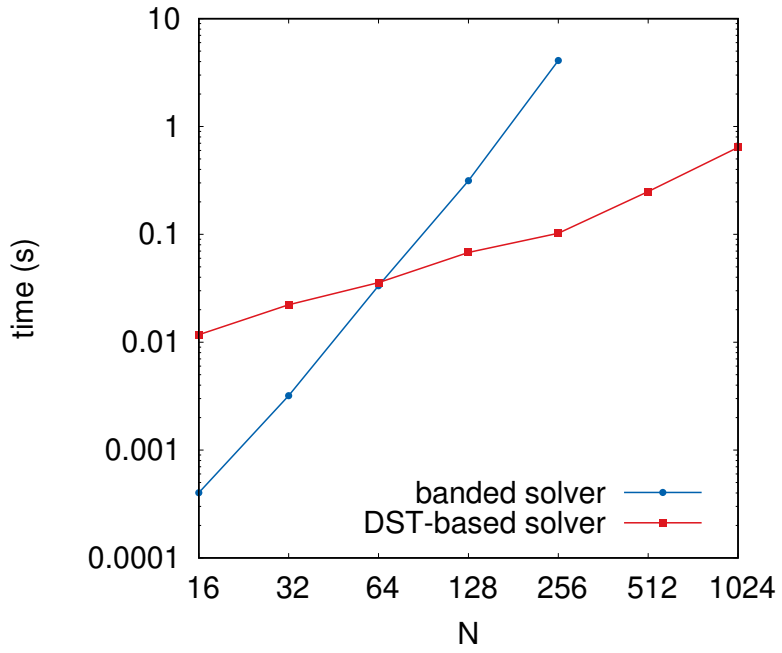


Figure 6: Timings for the banded and DST-based solvers.

(d) The iterative scheme utilizing the DST-based solver is

$$\nabla^2 u^{(i+1)} + k_0^2 u^{(i+1)} = v - \left(k(\mathbf{x})^2 - k_0^2\right) u^{(i)}, \tag{19}$$

with $u^{(0)} = 0$. The computed solutions are shown in Figures 7c and 7d for frequencies of $f = 21.3\,\text{MHz}$ and $f = 298.3\,\text{MHz}$, respectively. Convergence plots for both frequencies are shown in Figures 7e and 7f. The iterative method converges for the lower frequency but diverges for the higher frequency.

(e) Define the error at iteration i to be $\delta u^{(i)} = u^{(i)} - u$. Substituting $u^{(i+1)} = \delta u^{(i+1)} + u$ into Eq. 19 yields

$$\begin{aligned}
&\nabla^2 u^{(i+1)} + k_0^2 u^{(i+1)} = v - (k(\mathbf{x})^2 - k_0^2) u^{(i)} \\
\implies &(\nabla^2 + k_0^2) u^{(i+1)} = v - (k(\mathbf{x})^2 - k_0^2) u^{(i)} \\
\implies &(\nabla^2 + k_0^2)(\delta u^{(i+1)} + u) = v - (k(\mathbf{x})^2 - k_0^2) u^{(i)} \\
\implies &(\nabla^2 + k_0^2)\delta u^{(i+1)} + (\nabla^2 + k_0^2) u = v - (k(\mathbf{x})^2 - k_0^2) u^{(i)} \\
\implies &(\nabla^2 + k_0^2)\delta u^{(i+1)} = v - (k(\mathbf{x})^2 - k_0^2) u^{(i)} - (\nabla^2 + k_0^2) u
\end{aligned}$$

Since u satisfies the PDE, $\nabla^2 u = v - k(\mathbf{x})^2 u$. Substituting this in, we obtain

$$\begin{aligned}(\nabla^2 + k_0^2)\delta u^{(i+1)} &= v - (k(\mathbf{x})^2 - k_0^2)u^{(i)} - (\nabla^2 + k_0^2)u \\ &= v - (k(\mathbf{x})^2 - k_0^2)u^{(i)} - (v - k(\mathbf{x})^2 u + k_0^2 u) \\ &= (k_0^2 - k(\mathbf{x})^2)(u^{(i)} - u) \\ &= (k_0^2 - k(\mathbf{x})^2)\delta u^{(i)}\end{aligned}$$

Thus, we have

$$\delta u^{(i+1)} = (\nabla^2 + k_0^2)^{-1}(k_0^2 - k(\mathbf{x})^2)\delta u^{(i)}.$$

The iterative scheme will converge if the error is reduced at each iteration. We can rewrite the error at the $(i+1)$-th iteration in terms of the initial error as

$$\delta u^{(i+1)} = (\nabla^2 + k_0^2)^{-1}(k_0^2 - k(\mathbf{x})^2)^{i+1}\delta u^{(0)}.$$

The norm of the error at the $(i+1)$-th iteration is then bounded by

$$\begin{aligned}\|\delta u^{(i+1)}\| &= \|(\nabla^2 + k_0^2)^{-1}(k_0^2 - k(\mathbf{x})^2)^{i+1}\delta u^{(0)}\| \\ &\leq \|(\nabla^2 + k_0^2)^{-1}(k_0^2 - k(\mathbf{x})^2)\|^{i+1}\|\delta u^{(0)}\|.\end{aligned}$$

Thus, in order for the iterations to converge we need $\|(\nabla^2 + k_0^2)^{-1}(k_0^2 - k(\mathbf{x})^2)\| < 1$. This is true only when the spectral radius $\rho((\nabla^2 + k_0^2)^{-1}(k_0^2 - k(\mathbf{x})^2)) = |\lambda_{\max}| < 1$. Therefore, we need all eigenvalues of $(\nabla^2 + k_0^2)^{-1}(k_0^2 - k(\mathbf{x})^2)$ to have magnitude < 1. We can more explicitly determine this condition, since $(\nabla^2 + k_0^2)^{-1}$ is symmetric:

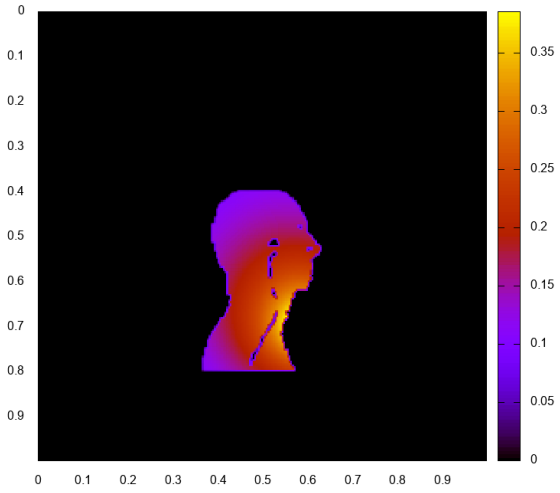
$$\begin{aligned}\|(\nabla^2 + k_0^2)^{-1}(k_0^2 - k(\mathbf{x})^2)\| &\leq \|(\nabla^2 + k_0^2)^{-1}\|\|k_0^2 - k(\mathbf{x})^2\| \\ &= \rho((\nabla^2 + k_0^2)^{-1})\|k_0^2 - k(\mathbf{x})^2\| \\ &= \max_j \left|\frac{1}{\lambda_j}\right| \|k_0^2 - k(\mathbf{x})^2\| \\ &= \frac{1}{\min_j |\lambda_j|}\|k_0^2 - k(\mathbf{x})^2\| \\ &< 1.\end{aligned}$$

where λ_j are the eigenvalues of $\nabla^2 + k_0^2$. This implies that

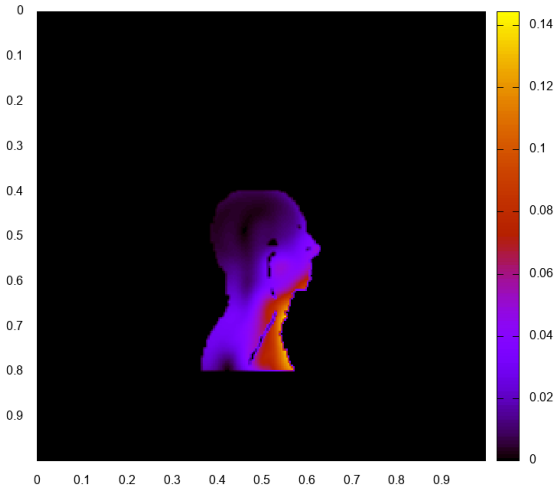
$$\|k_0^2 - k(\mathbf{x})^2\| < \min_j |\lambda_j| = \min_j \left| -\frac{2}{h^2}\left(1 - \cos\frac{j\pi}{N}\right) + \frac{k_0^2}{2}\right|.$$

So given a discretization spacing h we have a condition on $k(\mathbf{x})$ that guarantees convergence.

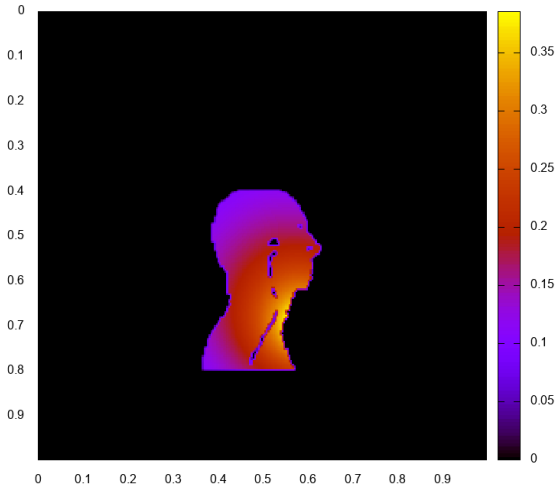
(f) It takes about 2 seconds for the iterative solver to converge to machine precision with $N = 1024$ and $f = 21.3\,\text{MHz}$. The scaling from Figure 6 indicates that it would take about 100 seconds for the banded solver to solve this same problem!



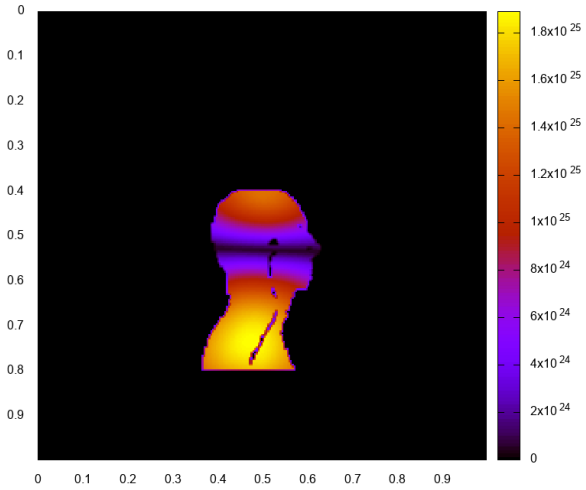
(a) Low frequency MRI solution using banded solver.



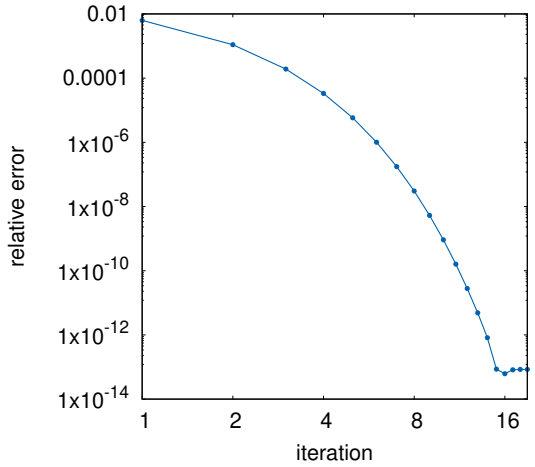
(b) High frequency MRI solution using banded solver.



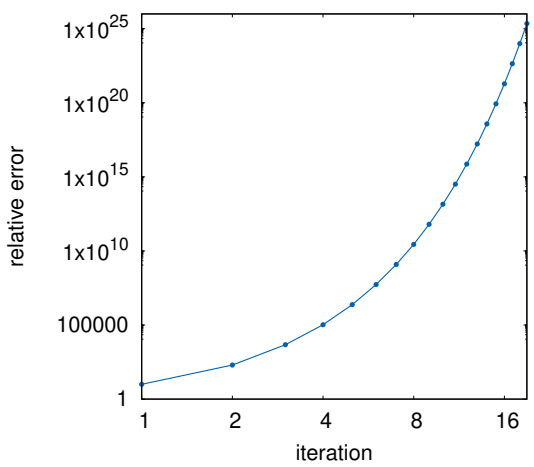
(c) Low frequency MRI solution using DST-based iterative method.



(d) High frequency MRI solution using DST-based iterative method.



(e) Low frequency convergence of iterative method.



(f) High frequency divergence of iterative method.

Figure 7

5. **Testing the Schur complement for the Poisson equation**

We wish to solve the Poisson equation $-\nabla^2 u = f$ on the domain $[0,1]^2$ with zero Dirichlet boundary conditions using (a) a single Poisson solve on a 112×112 grid using the FFT and (b) the Schur complement method on the perfect squared square (see Figure 8), with each subdomain solve using the FFT.

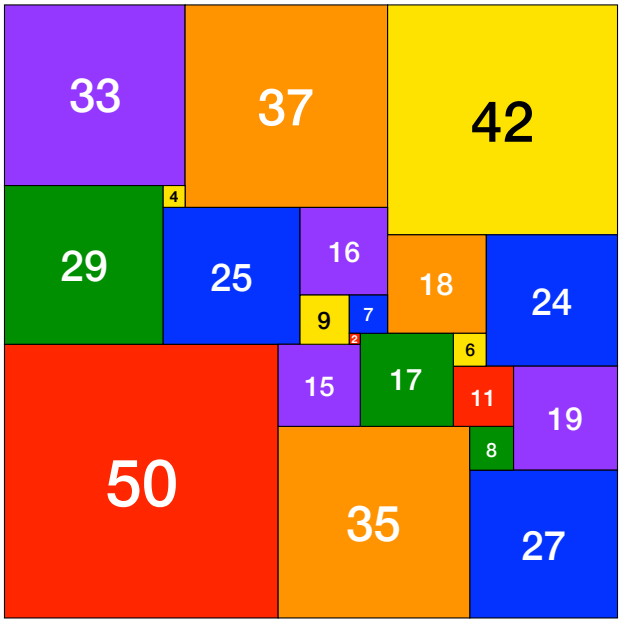


Figure 8: A perfect squared square.

The Poisson problem on the 21-subdomain perfect squared square leads to a linear system of the form

$$\begin{bmatrix} A_{1,1} & & & & A_{1,\Gamma} \\ & A_{2,2} & & & A_{2,\Gamma} \\ & & \ddots & & \vdots \\ & & & A_{21,21} & A_{21,\Gamma} \\ A_{\Gamma,1} & A_{\Gamma,2} & \cdots & A_{\Gamma,21} & A_{\Gamma,\Gamma} \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \\ \vdots \\ u_{21} \\ u_\Gamma \end{bmatrix} = \begin{bmatrix} f_1 \\ f_2 \\ \vdots \\ f_{21} \\ f_\Gamma \end{bmatrix}$$

where Γ indicates the gridpoints that lie on the interfaces or "glue" between the subdomains. Applying the Schur complement method to eliminate the degrees of freedom on the interiors of the subdomains results in an equation for the unknowns on the glue,

$$\Sigma u_\Gamma = f_\Gamma - \sum_{i=1}^{21} A_{\Gamma,i} A_{i,i}^{-1} f_i \tag{20}$$

where $\Sigma = A_{\Gamma,\Gamma} - \sum_{i=1}^{21} A_{\Gamma,i} A_{i,i}^{-1} A_{i,\Gamma}$ is called the Schur complement matrix.

Note that the matrix $A_{i,i}^{-1}$ applied to a vector corresponds to solving a Poisson problem on subdomain i, which can be done by calling our fast FFT-based Poisson solver. Additionally,

the matrix $A_{i,\Gamma}$ applied to a vector transfers information from the glue to the subdomain by injecting the given vector as Dirichlet data for the subdomain, and the matrix $A_{\Gamma,i}$ applied to a vector transfers information from the subdomain solution to the glue. Therefore, there is no need to form these matrices explicitly; if we use an iterative method such as conjugate gradient to solve Eq. 20, then it suffices to be able to multiply Σ by a vector.

The solution to the interface problem Eq. 20 is shown in Figure 9d. Note that Eq. 20 only gives us the computed solution on the glue; it remains to compute the solution on the subdomains.

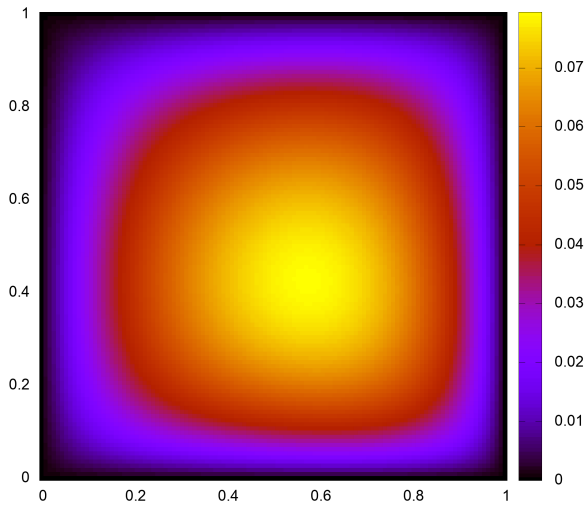
Once we have solved Eq. 20, we can compute the solutions on the subdomains by solving the decoupled Poisson problems

$$A_{i,i}u_i = f_i - A_{i,\Gamma}u_\Gamma$$

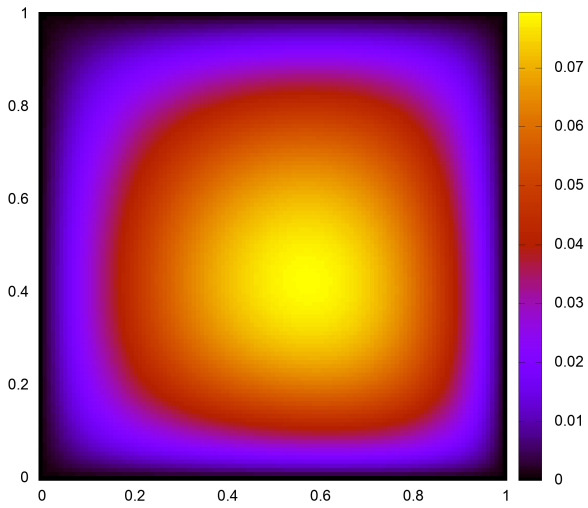
for $i = 1, 2, \ldots, 21$.

The solutions computed by methods (a) and (b) with $f(x, y) = e^{x-y}$ are shown in Figures 9a and 9b, respectively. The error between the two solutions is shown in Figure 9c. The maximum norm error is 8.46545×10^{-16}, indicating that the two solutions agree up to machine precision.

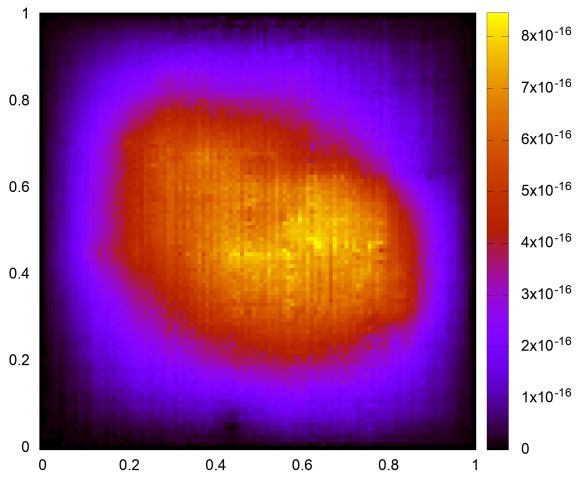
The trickiest aspect of this problem is the indexing, which determines how the glue talks to the subdomains and vice versa. In the provided code, the relationships between the glue and the subdomains are computed once and stored in a dictionary structure (`std::unordered_map`). This makes the indexing much easier, as the communication between the glue and subdomains amounts to looking up the indices in the dictionary.



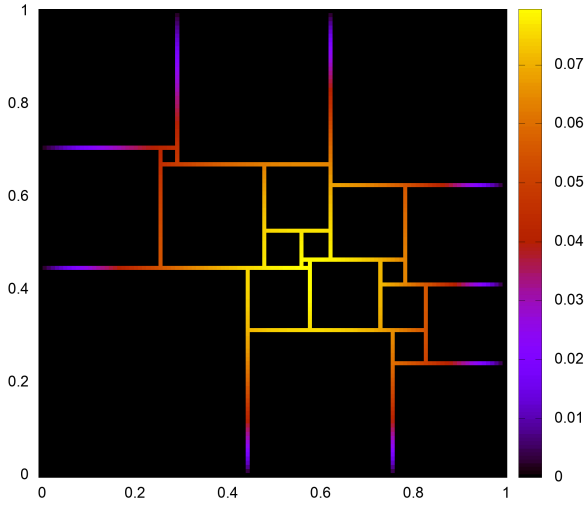
(a) The solution on the single domain.



(b) The solution using the Schur complement method.



(c) The error between the computed solutions.



(d) The solution on the interface.

Figure 9