

Math 471 Project 3 heat

Erik Erhardt

December 5, 2005

1. Problem Description. *Marshak wave*

The goal of this project is to simulate the propagation of a Marshak wave through the implementation of a finite difference/ghost cell based Laplace operator in parallel.

2. Laplacian routine maximum error. Table 1 on page 1 shows three decomposition methods resulting in the same maximum error, giving evidence that the Laplacian routine is working correctly in my code.

| Global size | Decomposition dimensions | | |
|-------------|--------------------------|--------------|--------------|
| | 1x1 | 5x5 | 1x25 |
| 100x100 | 0.0015910596 | 0.0015910596 | 0.0015910596 |
| 200x200 | 0.0004018149 | 0.0004018149 | 0.0004018149 |
| 400x400 | 0.0001009606 | 0.0001009606 | 0.0001009606 |
| 800x800 | 0.0000253035 | 0.0000253035 | 0.0000253035 |

Table 1: Laplacian routine maximum error

The plot in Figure 1 on page 1 shows the three decomposition methods to be equivalent.

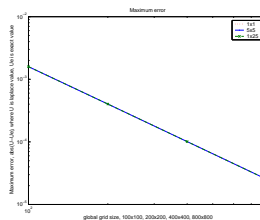


Figure 1: Log-Log plot of the maximum error of 3 decomposition methods and four global grid sizes.

3. CFL condition. It appears that $h = \Delta t < (\Delta x)^2/8$ is a stable time step. I set $h = \frac{1}{4}(\Delta x)^2/8$ to assure stability.

4. Scalability. Scalability plots are presented in Figure 2 on page 2. It appears the greatest benefit is achieved with 4 processors on a single node. Beyond the single machine, the burden of internode communication causes time per timestep to increase. Performance is better for $a \times a$ decomposition, than for $1 \times a$. Similar results are for global grid sizes of 200 and 400, except that for 400, where there is proportionally more computation than communication than for 200, better performance occurs for more processes. Table 2 on page 2 gives the values plotted.

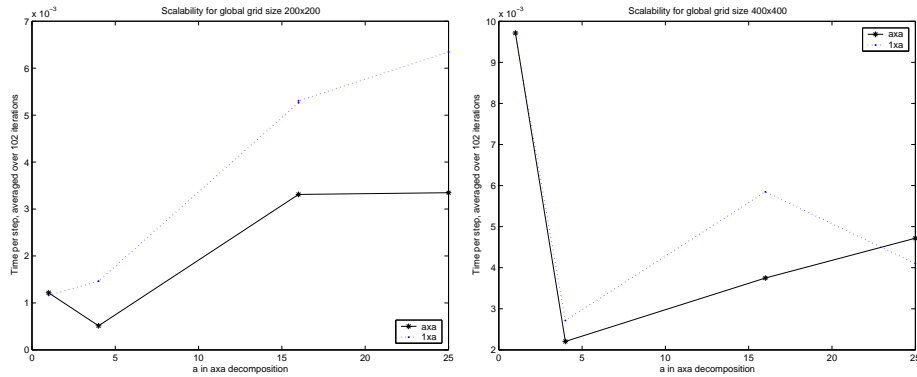


Figure 2: Scalability plot.

| 200 | | | | 400 | | | |
|-----|----------|------|----------|-----|----------|------|----------|
| 1x1 | 0.001216 | 1x1 | 0.001163 | 1x1 | 0.009714 | 1x1 | 0.009675 |
| 2x2 | 0.000513 | 1x4 | 0.001463 | 2x2 | 0.002205 | 1x4 | 0.002714 |
| 4x4 | 0.003311 | 1x16 | 0.005265 | 4x4 | 0.003749 | 1x16 | 0.005846 |
| 5x5 | 0.003347 | 1x25 | 0.006346 | 5x5 | 0.004718 | 1x25 | 0.004102 |

Table 2: Scalability: time per timestep for both decomposition methods. (note: 1x16 200 uses 192 grid size)

5. Contour plot of solution using 200x200. I used 200x200 for time considerations. In Figure 3 on page 3 are 11 plots, for $t = 0, t = 0.05, \dots, t = 0.50$. These plots are very satisfying.

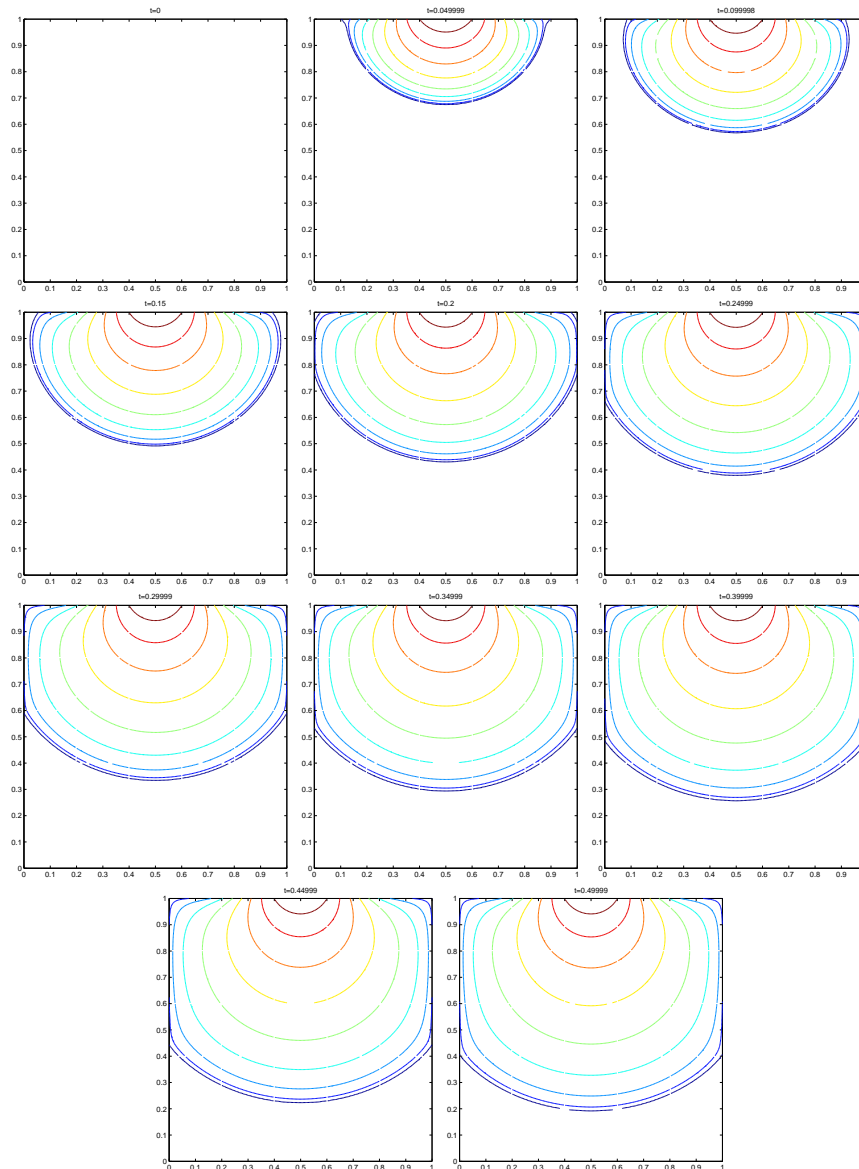


Figure 3: Contour plots of heat propagation.

Appendix

Matlab code for plots

```

% laplacian absolute error
x=[
100 0.0015910596 0.0015910596 0.0015910596
200 0.0004018149 0.0004018149 0.0004018149
400 0.0001009606 0.0001009606 0.0001009606
800 0.0000253035 0.0000253035 0.0000253035
]
loglog(x(1,:),x(2,:), 'r-', x(1,:),x(3,:), '.-', x(1,:),x(4,:), 'x--')
legend('1x1', '5x5', '1x25', 0);
title('Maximum error');
xlabel('global grid size, 100x100, 200x200, 400x400, 800x800');
ylabel('Maximum error, abs(U-Ue), where U is laplace value, Ue is exact value');

% scalability results
% size nx ny time timeperstep
x=[
200 1 1 1 0.124068 0.001216
200 4 2 2 0.052376 0.000513
200 16 4 4 0.337709 0.003311
200 25 5 5 0.341415 0.003347
]
y=[
200 1 1 1 0.118664 0.001163
200 4 1 4 0.149202 0.001463
192 16 1 16 0.537060 0.005265
208 16 1 16 0.541248 0.005306
200 25 1 25 0.647309 0.006346
]
plot(x(2,:),x(6,:), 'k*-', y(2,:), y(6,:), 'b.:')
legend('axa', '1xa', 0);
title('Scalability for global grid size 200x200');
xlabel('a in axa decomposition');
ylabel('Time per step, averaged over 102 iterations');

x=[
400 1 1 1 0.990878 0.009714
400 4 2 2 0.224908 0.002205
400 16 4 4 0.382369 0.003749
400 25 5 5 0.481191 0.004718
]
y=[
400 1 1 1 0.986863 0.009675
400 4 1 4 0.276812 0.002714
400 16 1 16 0.596339 0.005846
400 25 1 25 0.418443 0.004102
]
plot(x(2,:),x(6,:), 'k*-', y(2,:), y(6,:), 'b.:')
legend('axa', '1xa', 0);
title('Scalability for global grid size 400x400');
xlabel('a in axa decomposition');
ylabel('Time per step, averaged over 102 iterations');

```