Solving 2-dimensional time-dependent heat-conduction equations with source on square lattice using v-cycle and w-cycle multigrid methods with MPI acceleration

Chunshu Wu
happycwu@bu.edu

Leyang Yu
yly@bu.edu

Shidong Sun
Shidongs@bu.edu

April 10, 2019

Abstract

In this paper, three different types of 2-dimensional time-dependent heat conduction equations are solved. Non-zero initial heat field without source, zero initial heat field with a constant source, and zero initial heat field with a source that varies with time. The equations are all solved on square lattice with boundaries fixed at zero. Two multigrid methods, v-cycle and w-cycle multigrid are used to solve the equations. For further performance improvement, we accelerate our program with MPI for multi-processors.

The v-cycle and w-cycle methods are compared time-wise, and the speedup of multi-processors is measured. In order to intuitively show how the heat is conducted, the solutions are made into videos.

1 Mathematical Background

The equations to be solved are generally of the shape as below

$$\frac{\partial u}{\partial t} - \frac{\partial^2 u}{\partial x^2} - \frac{\partial^2 u}{\partial y^2} = f(t, x, y) \tag{1}$$

Where u is the heat value to be solved for, t is the unscaled time, x and y are unscaled spacial variables, and f is the source term determined arbitrarily.

1.1 Discretize the equation

The 2nd-order derivatives are normally discretized with both forward finite difference and backward finite difference.

$$\frac{\partial^2 u}{\partial x^2} \xrightarrow{\text{finite difference}} \frac{u[ih+h]+u[ih-h]-2u[ih]}{h^2} \tag{2}$$

Where h is the smallest inseparable unit that x can be divided into and i is the index on x direction. In computers, this formula is discretized with $h = 1$. Therefore,

$$\frac{\partial^2 u}{\partial x^2} \xrightarrow{\text{discretize}} u[i+1]+u[i-1]-2u[i] \tag{3}$$

Take the y components into consideration,

$$\frac{\partial^2 u}{\partial x^2}+\frac{\partial^2 u}{\partial y^2} \xrightarrow{\text{discretize}} u[i+1,j]+u[i-1,j]+u[i,j+1]+u[i,j-1]-4u[i,j] \tag{4}$$

The scale is the same for both x and y directions because the equations are solved on square lattice. However, in the case of time, the finite difference is not necessarily the same as space.

$$\frac{\partial u}{\partial t} \xrightarrow{\text{discretize}} \frac{u[t]-u[t-g]}{g} \tag{5}$$

g is the stride of time. Here backward finite time difference method is used, because there will be fluctuation if we use forward finite time difference[1]. Sum the above up, the differential equation becomes

$$\begin{split}\frac{u[t,i,j]-u[t-g,i,j]}{g}+4u[t,i,j]-u[t,i+1,j]-u[t,i-1,j]\\ -u[t,i,j+1]-u[t,i,j-1]=f[t,i,j]\end{split} \tag{6}$$

1.2 Iteration Equation

Rearrange the terms from equation (6) and the iteration equation is obtained

$$\begin{split}u[t,i,j]=\frac{g}{4g+1}(u[t,i+1,j]+u[t,i-1,j]+u[t,i,j+1]+u[t,i,j-1])\\ +\frac{1}{4g+1}u[t-g,i,j]+\frac{g}{4g+1}f[t,i,j]\end{split} \tag{7}$$

It's worth noting that if g approaches infinity, the $u[t-g, i, j]$ term is vanished and the equation becomes the time-independent equilibrium equation. If g approaches 0, $u[t, i, j]$ becomes the same as $u[t-g, i, j]$.

2 Multigrid

The idea of multigrid is to separate the exact solution and error, and use a layered structure to accelerate the convergence.
In our case, the exact solution is separated as below

$$u[t, i, j]_{exact} = u[t, i, j] + e[t, i, j] \tag{8}$$

u[t,i,j] is the approximate solution and e[t,i,j] is the error. The residue is

$$r[t, i, j] = f[t, i, j] - A^{t,i,j}_{t',i',j'} u[t', i', j'] \tag{9}$$

Substitute equation (8) into equation (9), with the fact that

$$A^{t,i,j}_{t',i',j'} u[t', i', j']_{exact} = f[t, i, j] \tag{10}$$

We obtain

$$r[t, i, j] = A^{t,i,j}_{t',i',j'} u[t', i', j'] e[t', i', j'] \tag{11}$$

Hence the error can be solved using equation (11). If the error is considered as the solution of equation (11), there is error for the error as well. Therefore, the errors can be put in layers of different significance. As the layer gets deeper, the error becomes less significant. In that case, the deeper layers can be coarser without much loss in accuracy eventually. Once an error is approximately calculated, the corresponding solution gets corrected, and the correction is propagated to the layers above.

There are many varieties of multigrid cycle architectures, v-cycle, w-cycle, f-cycle[2], etc. In this paper, we will use and compare two typical architectures, v-cycle and w-cycle.

2.1 V-cycle

The multigrid method can be divided into three components. Relaxation, residue projection, and error interpolation.

References

[1] Gerald W. Recktenwald. *Finite difference Approximations to the Heat Equation (previous project)*. 2011.

[2] A.Schüller U.Trottenberg C.W.Oosterlee. *Multigrid*. Academic Press, 2001.