2D Problem

$$\Delta u(x, y) + k^2(y) u(x, y) = f(x, y)$$

Let N_x and N_y be the number of grid points in the $x-$ and $y-$ directions respectively. Then discretization leads to the block matrix equation

$$\begin{bmatrix} A & C & & & \\ B & A & C & & \\ & \ddots & \ddots & \ddots & \\ & & B & A & C \\ & & & B & A \end{bmatrix} \begin{bmatrix} U_1 \\ U_2 \\ \vdots \\ U_{N_y-1} \\ U_{N_y} \end{bmatrix} = \begin{bmatrix} F_1 \\ F_2 \\ \vdots \\ F_{N_y-1} \\ F_{N_y} \end{bmatrix}$$

where $A, B, C \in \mathbb{C}^{N_x \times N_x}$ are tridiagonal matrices.

Solver

System above multiplied out:

$$AU_1 + CU_2 = F_1$$

$$BU_{j-1} + AU_j + CU_{j+1} = F_j \quad j = 2, \ldots, N_y - 1$$

$$BU_{N_y-1} + AU_{N_y} = F_{N_y}$$

Transform the system:

$$V^T AVV^T U_1 + V^T CVV^T U_2 = V^T F_1$$

$$V^T BU_{j-1} + V^T AVV^T U_j + V^T CVV^T U_{j+1} = V^T F_j \quad j = 2, \ldots, N_y - 1$$

$$V^T BVV^T U_{N_y-1} + V^T AVV^T U_{N_y} = V^T F_{N_y}$$

where V is the scaled matrix of eigenvectors such that $VV^T = I$.

The result is a tridiagonal system:

$$\Lambda_A \bar{U}_1 + \Lambda_C \bar{U}_2 = V^T F_1$$

$$\Lambda_B \bar{U}_{j-1} + \Lambda_A \bar{U}_j + \Lambda_C \bar{U}_{j+1} = V^T F_j \qquad j = 2, \ldots, N_y - 1$$

$$\Lambda_B \bar{U}_{N_y-1} + \Lambda_A \bar{U}_{N_y} = V^T F_{N_y}$$

where Λ_A, Λ_B,Λ_C are diagonal matrices of eigenvalues. Here $V^T F_j$ is the DST of F_j. This DST is found by taking computing the

Algorithm 1 Sequential 2D Solver

for $j = 1, \ldots, N_y$ **do**
    1D forward DST of F_j
end for
for $i = 1, \ldots, N_x; j = 1, \ldots, N_y$ **do**
    Solve the tridiagonal system for $\bar{U}_j$ using LU decomposition
end for
for $j = 1, \ldots, N_y$ **do**
    1D reverse DST of $\bar{U}_j$
end for

Note that the LU decomposition is calculated previously using the known eigeinvalues.

Calculating the DST via FFT

Let $E_j \in \mathbb{C}^{2N_x+2}$ be such that

$$E_j = \begin{bmatrix} 0 & F_{1,j} & \dots & F_{N_x,j} & 0 & \dots & 0 \end{bmatrix}^T$$

Then

$$V^T F_j = DST(F_j) = -Im\left(FFT\left(E_j\right)\right)\Big|_2^{N_x+1}$$

Hackathon Focus

1. Move the computation of the solution on a GPU in 2D
    - Forward FFT
    - Tridiagonal Solver
    - Reverse FFT
2. Use multiple GPUs
3. Extend to the 3D problem

C Implementation

int main (int argc, char **argv) {

defineSystem(argc, argv) [set values, contained in struct 'sys']

coefficients(sys) [set stencil coefficients, 2nd, 4th and 6th order]

rhs(sys) [set rhs (F), 2nd, 4th and 6th order]

LU(sys) [compute LU decomposition of the tridiagonal matrix]

solver(sys) [solver described above]

residual(sys) [calculate residual to check convergence]

clearMemory(sys) [clear allocated memory]

}