Analog Computing for Linear Systems

Yipeng Huang
December 15, 2015
A Simple Linear ODE and Digital Solution...

\[ \frac{d\vec{u}}{dt} = A\vec{u}(t) + \vec{b} \]

\[ \frac{d}{dt} \begin{bmatrix} u_0 \\ u_1 \end{bmatrix} = \begin{bmatrix} a_{00} & a_{01} \\ a_{10} & a_{11} \end{bmatrix} \begin{bmatrix} u_0(t) \\ u_1(t) \end{bmatrix} + \begin{bmatrix} b_0 \\ b_1 \end{bmatrix} \]

With initial conditions:

\[ \vec{u}(0) = u_{init} \]
A Simple Linear ODE and Digital Solution...

\[
\frac{du}{dt} = A\vec{u}(t) + \vec{b}
\]

\[
\frac{d}{dt} \begin{bmatrix} u_0 \\ u_1 \end{bmatrix} = \begin{bmatrix} a_{00} & a_{01} \\ a_{10} & a_{11} \end{bmatrix} \begin{bmatrix} u_0(t) \\ u_1(t) \end{bmatrix} + \begin{bmatrix} b_0 \\ b_1 \end{bmatrix}
\]

With initial conditions:

\[
\vec{u}(0) = \vec{u}_{init}
\]

```c
double time = 1.0;
int steps = 100;
double step_size = time / (double)steps;
for (int step=0; step<steps; step++) {
    delta_0 = b_0 + A_00 * u.curr0 + A_01 * u.curr1;
    delta_0 *= step_size;
    u.curr0 += delta_0;
    delta_1 = b_1 + A_10 * u.curr0 + A_11 * u.curr1;
    delta_1 *= step_size;
    u.curr1 += delta_1;
}
```
...and Equivalent Analog Solution

\[
\frac{d\vec{u}}{dt} = A\vec{u}(t) + \vec{b}
\]

\[
\frac{d}{dt} \begin{bmatrix} u_0 \\ u_1 \end{bmatrix} = \begin{bmatrix} a_{00} & a_{01} \\ a_{10} & a_{11} \end{bmatrix} \begin{bmatrix} u_0(t) \\ u_1(t) \end{bmatrix} + \begin{bmatrix} b_0 \\ b_1 \end{bmatrix}
\]

With initial conditions:

\[ \vec{u}(0) = \vec{u}_{init} \]
Physical Hardware
A Simple Linear PDE

\[ \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = b(x, y) \]
A Simple Linear PDE

\[ \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = b(x, y) \]

\[
\begin{array}{ccc}
  u_0 & u_1 & u_2 \\
  u_3 & u_4 & u_5 \\
  u_6 & u_7 & u_8 \\
\end{array}
\]

\[
A = \frac{1}{3^2} \begin{bmatrix}
  4 & -1 & -1 \\
 -1 & 4 & -1 \\
 -1 & -1 & 4 \\
 -1 & 4 & -1 \\
 -1 & -1 & -1 \\
 -1 & 4 & -1 \\
 -1 & -1 & 4 \\
 -1 & 4 & -1 \\
 -1 & -1 & 4 \\
\end{bmatrix}
\]

\[ Au = b \]

\[
\mathbf{u} = [u_0, u_1, \ldots, u_8]^{\top}, \quad \mathbf{b} = [b_0, b_1, \ldots, b_8]^{\top}
\]
Taxonomy of PDE / ODE, Grid / Stencil Problems

- Partial differential equation (PDE)
  - Time dependent PDE
    - Parabolic PDE (e.g., heat equation)
      - Spatial discretization (e.g., finite difference)
      - System of ordinary differential equations (ODE)
        - Explicit time stepping (e.g., RK4, analog)
        - Implicit time stepping (e.g., backward Euler)
          - Sparse system of linear equations (SLE)
            - Iterative solvers (e.g., CG, analog)
  - Time independent PDE
    - Hyperbolic PDE (e.g., wave equation)
      - Spatial discretization (e.g., finite difference)
    - Elliptic PDE (e.g., Poisson eq.)
      - Spatial discretization (e.g., finite difference)
      - Nonlinear system of equations
        - Nonlinear solvers (e.g., Newton’s)

- Dense linear eqs. (e.g., optimization)
  - Direct solvers (e.g., Cholesky, QR, SVD)
Overview of Approach and Findings

• Use analog computing as a “super operator” for sparse problems
  • Then, using multigrid, domain decomposition, and other higher level algorithms, analog can help solve larger problems
Overview of Approach and Findings

• Use analog computing as a “super operator” for sparse problems
  • Then, using multigrid, domain decomposition, and other higher level algorithms, analog can help solve larger problems

• Findings
  • Analog is fast, but not game-changing; Digital has advantage in CG algorithm
  • Analog area consumption is high

• What it needs to succeed
  • Optimum gradient method, discussed in in Karplus and Korn
  • I’ll do in depth experiments using HCDC v2
  • Vary problem size (up to 4x4), 2d, 3d, dense connectivity
  • Randomized coefficients and boundary conditions
Performance Comparison

![Graph showing performance comparison between CG convergence time and analog convergence time.](image-url)
Performance Comparison

![Graph comparing convergence time with total grid points for digital CG, analog 20KHz, and linear (analog 80KHz projection).](image)
Design Space Exploration

![Graph showing convergence time versus total grid points for different frequencies: digital CG, analog 20KHz, linear projection for 20KHz, linear projection for 80KHz, linear projection for 320KHz, linear projection for 1.3MHz.](image)
Energy

![Energy Graph]

- Solution energy (µJ) vs. total grid points
- Lines represent different frequencies:
  - GPU
  - 20 KHz
  - 80 KHz
  - 320 KHz
  - 1.3 MHz
Next Directions for Analog

• Dense linear algebra

• Nonlinear systems of equations

• Stochastic simulation