Matrix Multiplication and Partial Differential Equations via Wavelets

by

Daniel D. Beatty
Computer Science
Texas Tech University

Topics of Discussion

- An Overview on Wavelets
  - Wavelet Transform Definition
  - 1-D Wavelets and Convolution
  - 2-D Wavelets and Multi-Resolution on the average term
  - 2-D Wavelets, Multi-Resolution Quad-Trees, and their optimization
- Matrix Multiplication and its Wavelet Version
  - Proof
  - Chain Structure
- Partial Differential Equations
  - Overview of PDE
  - Related Work
  - Proposed Work

Why Is This Important

- Optimized Computations
  - Well Conditioned Matrices
  - Sparse Matrices
- Application of Matrix Multiplication
  - Simulations, Computer Vision
  - Geographical Information Systems (GIS)
- Application of Partial Differential Equations
  - Gas Flow Simulations
  - Particle Physics Simulations

Thesis Objectives

- Produce wavelet filter library
- Apply wavelet operator in matrix multiplication
  - Determine levels approximation and accuracy
  - Determine level of efficiency
- Apply wavelet operator to partial differential equations
  - Determine levels approximation and accuracy
  - Determine level of efficiency

Overview on Wavelets

- Wavelet Transform Definition
- 1-D Wavelets and Convolution
- 2-D Wavelets and Multi-Resolution on the average term
- 2-D Wavelets, Multi-Resolution Quad-Trees, and their optimization

Definition of Wavelet

- A wavelet vector/ array is defined as a generalization of an even length array by an orthonormal bases which are subsets of the entire array.
- A wavelet matrix is defined as a generalization of a square orthogonal or unitary matrix which is a subset of a large class of rectangular matrices

Array Wavelet Transform

- Array (1-D) Wavelets are defined in terms of average and difference terms.
- The average term is an array half the size of the array it represents.
- The difference term is an array half the size of the array it represents.
- Typically the mapping of the wavelet transform is defined simply as $S \rightarrow (A|D)$.
- Each term is acquired by convolving S with some filter. These filters are chosen based on the properties as a pair. Orthonormality, compactness, and smoothness are just a few properties.

Convolution

- Convolution is an operation which applies two functions together. Defined mathematically: $x * h = \sum_l h(l)x(k-l)$
- The algorithm is as follows:

$\forall i \in [0, M)$
  $\forall j \in [0, N)$
    `n=i-j`
    `if` $(n \in [0, M))$
      $y_i += x_n \cdot h_j$

Convolution applied to the Wavelet Transform

$$A_i = A_{i-1} * V, \qquad D_i = A_{i-1} * W$$

where

- V is the scaling wavelet vector,
- W is the differencing wavelet vector,
- A is the average vector (scaled vector),
- D is the difference component vector,
- $\forall i \in [1, L)$ and $A_0 = f$ which is the original signal, and
- L is the limit on the number resolutions that signal can have based on the wavelet type.

Odd-versus-Even Indexing

- There is a significant issue as to how the difference terms, and average terms are indexed.
- The issue is how each element is accounted for in the wavelet representation.

$$(S_i, S_{i-1}) \to A_i, \text{ and } (S_i, S_{i-1}) \to D_i.$$

- The following equations represent the mapping from the average and difference arrays to the recovered array.

$$\begin{array}{rll} \text{Odd:} & R_{2i} = (A_i - D_i)\sqrt{1/2}, & R_{2i+1} = (A_i + D_i)\sqrt{1/2} \\ \text{Even:} & R_{2i} = (A_i + D_i)\sqrt{1/2}, & R_{2i-1} = (A_i - D_i)\sqrt{1/2} \end{array}$$

2-D Wavelet Transform

- The 2D transform has four components:
  - A: the average (average of the rows and columns)
  - V: vertical (average of the columns and difference of the rows)
  - H: horizontal (average of the rows and difference of the columns)
  - D: diagonal (difference of the rows and columns)

$$B \Rightarrow \begin{pmatrix} H & D \\ A & V \end{pmatrix} \qquad B \Rightarrow \begin{pmatrix} A & V \\ H & D \end{pmatrix}$$

1-D to 2-D Method

- Given:
  - 1D wavelet transform source matrix
  - Wavelet basis
- Solution: $\forall i \in rows$
  - $\forall j \in columns$
  - $S[j] \leftarrow source[i][j]$
  - $S \Rightarrow^{W} R$

Vector-Matrix Method

Row Transform

$\forall i \in rows$

1. Initialize temporary array/vector to all zeros (x).
2. $\forall k \in columns$

   (a) $\forall l \in ha.Size\ x+ = \begin{cases} S_{i,k-l} * hA_l, & \text{if } k-l \in columns \\ 0, & \text{otherwise} \end{cases}$

3. $\forall k \in columns/2$

   $result_{i,k} = x_{2k+1}$ (In other words, odd split)

4. Initialize x to all zeros.
5. $\forall k \in columns$

   (a) $\forall l \in hd.Size\ x+ = \begin{cases} S_{i,k-l} * hD_l, & \text{if } k-l \in columns \\ 0, & \text{otherwise} \end{cases}$

6. $\forall k \in columns/2$

   $result_{i,k+columns/2} = x_{2k+1}$ (In other words, odd split)

Multi-Resolution

1. Initialize xA and xD to zero
2. $\forall k \in columns, \qquad \forall l \in filter$
   - $n = k - l$
   - if ($n \in columns$)
     $xA_k = W_{i,n} * hA_l$
     $xD_k = W_{i,n} * hD_l$
3. Transfer back to W
   $W_i = xA|xD$

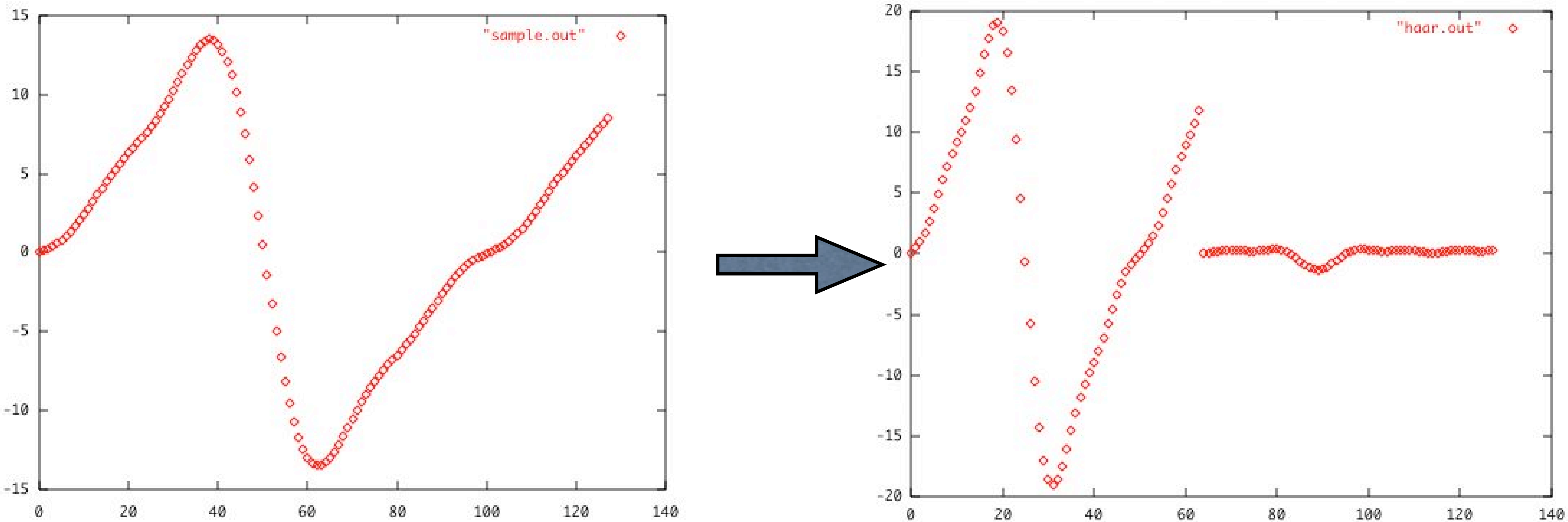
And the column transform is represented by:

1. initialize yA and yD to zero
2. $\forall k \in rows, \qquad \forall l \in filter$
   - $n = k - l$
   - if ($n \in columns$)
     $yA_k = W_{i,n} * hA_l$
     $yD_k = W_{i,n} * hD_l$
3. Transfer back to W
   $W_j = yA|yD$

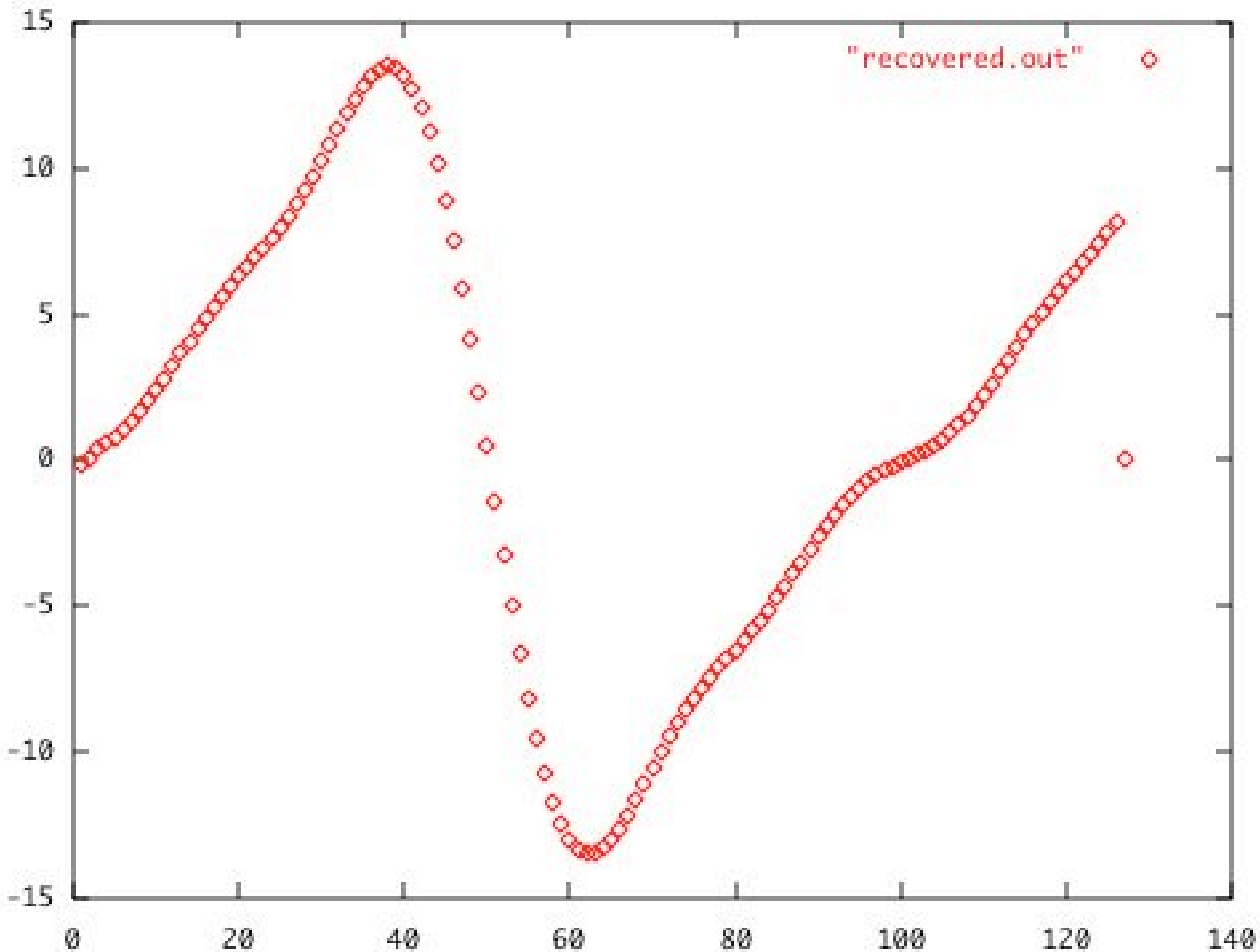
Proof of Concept

- 1-D Transform
- 2-D Transform
- 2-D Multi-Resolution
- 2-D Multi-Resolution Quad Tree

1-D Transform



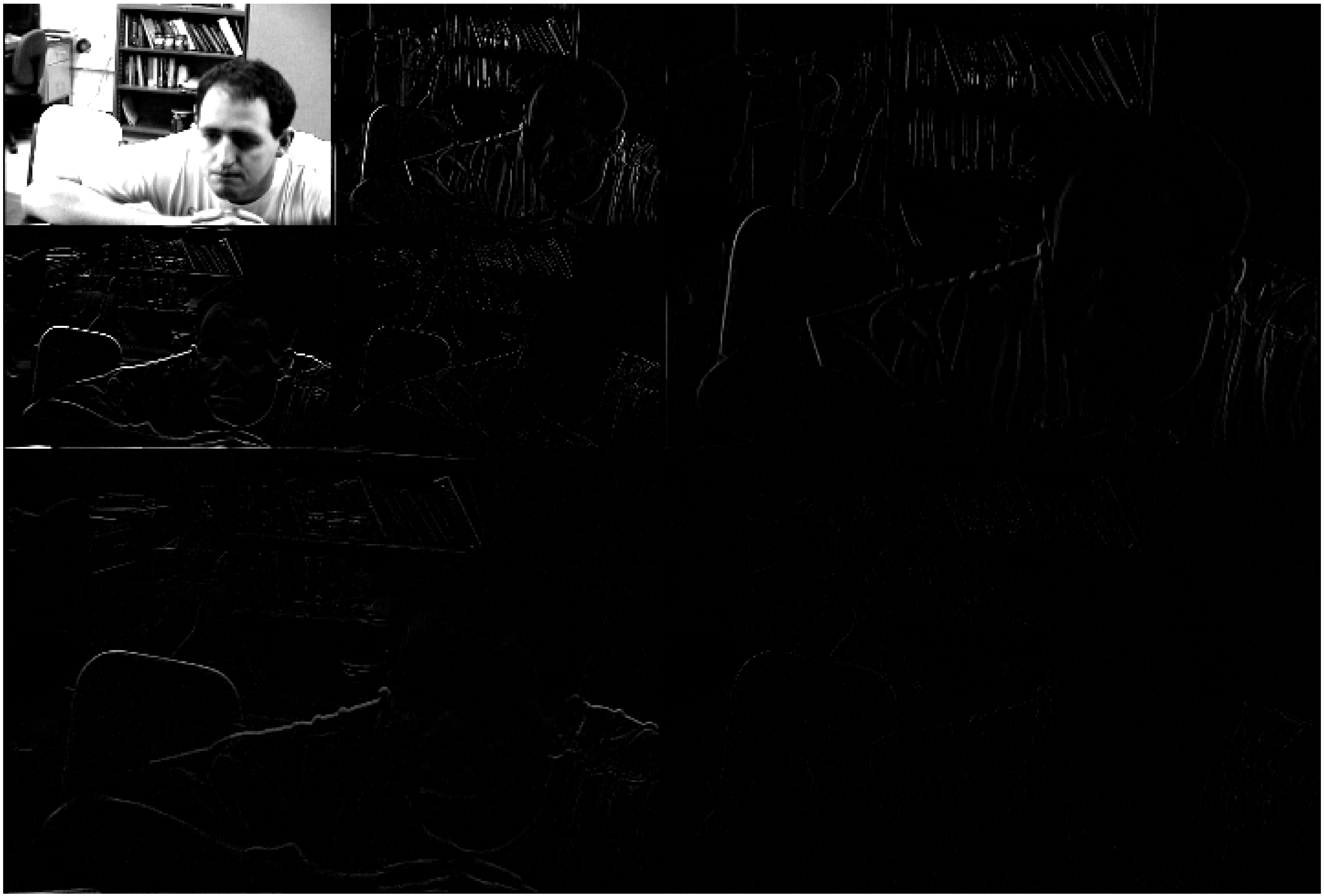
1-D Transform Even Index



2-D Original Image

2-D Wavelet Transform

2-D Multi-Resolution



Matrix Multiplication

- Concept
  - Proof
  - 2×2 example
  - Qualifications for Wavelets in Matrix Multiplication
  - Wavelet Based Matrix Multiply Procedure
- The objectives of this thesis
  - Apply wavelet operator in matrix multiplication
  - determine levels approximation/ accuracy
  - determine efficiency

Theoretical Proof

This segment is a simple linear algebra proof which shows that wavelet matrix multiplication is possible.

1. $L(AB) = L(A) \cdot L(B)$
2. ψ^{-1} is a linear operator
3. $\psi^{-1}(\psi(A) \cdot \psi(B)) = \psi^{-1}(\psi(A)) \cdot \psi^{-1}(\psi(B))$
4. $\psi^{-1}(\psi(A)) = A$
5. $\psi^{-1}(\psi(B)) = B$
6. Therefore: $AB = \psi^{-1}(\psi(A) \cdot \psi(B))$

A 2 by 2 Example

$$c_i^j = \sum_k a_i^k \ b_k^j.$$

$$\begin{pmatrix} a_1^1 & a_1^2 \\ a_2^1 & a_2^2 \end{pmatrix} \begin{pmatrix} b_1^1 & b_1^2 \\ b_2^1 & b_2^2 \end{pmatrix} = \begin{pmatrix} a_1^1 b_1^1 + a_1^2 b_2^1 & a_1^1 b_1^2 + a_1^2 b_2^2 \\ a_2^1 b_1^1 + a_2^2 b_2^1 & a_2^1 b_1^2 + a_2^2 b_2^2 \end{pmatrix}$$

$$W(A) = \frac{1}{2} \begin{pmatrix} (a_1^1 + a_2^1 + a_1^2 + a_2^2) & (a_1^1 + a_2^1 - a_1^2 - a_2^2) \\ (a_1^1 - a_2^1 + a_1^2 - a_2^2) & (a_1^1 - a_2^1 - a_1^2 + a_2^2) \end{pmatrix}, \tag{1}$$

$$W(B) = \frac{1}{2} \begin{pmatrix} (b_1^1 + b_2^1 + b_1^2 + b_2^2) & (b_1^1 + b_2^1 - b_1^2 - b_2^2) \\ (b_1^1 - b_2^1 + b_1^2 - b_2^2) & (b_1^1 - b_2^1 - b_1^2 + b_2^2) \end{pmatrix}. \tag{1}$$

$$A \cdot B = \begin{pmatrix} (a_1^1 b_1^1 + a_1^2 b_2^1) & (a_1^1 b_1^2 + a_1^2 b_2^2) \\ (a_2^1 b_1^1 + a_2^2 b_2^1) & (a_2^1 b_1^2 + a_2^2 b_2^2) \end{pmatrix}.$$

$$
\begin{aligned}
\psi(A) &= (a_1^1 b_1^1 + a_1^2 b_2^1 + a_1^1 b_1^2 + a_1^2 b_2^2) + (a_2^1 b_1^1 + a_2^2 b_2^1 + a_2^1 b_1^2 + a_2^2 b_2^2) \\
\psi(V) &= (a_1^1 b_1^1 + a_1^2 b_2^1 - a_1^1 b_1^2 - a_1^2 b_2^2) + (a_2^1 b_1^1 + a_2^2 b_2^1 - a_2^1 b_1^2 - a_2^2 b_2^2) \\
\psi(H) &= (a_1^1 b_1^1 + a_1^2 b_2^1 + a_1^1 b_1^2 + a_1^2 b_2^2) - (a_2^1 b_1^1 + a_2^2 b_2^1 + a_2^1 b_1^2 + a_2^2 b_2^2) \\
\psi(D) &= (a_1^1 b_1^1 + a_1^2 b_2^1 - a_1^1 b_1^2 - a_1^2 b_2^2) - (a_2^1 b_1^1 + a_2^2 b_2^1 - a_2^1 b_1^2 - a_2^2 b_2^2)
\end{aligned}
$$

$$
\begin{aligned}
\psi(A) &= a_1^1 b_1^1 + a_1^2 b_2^1 + a_1^1 b_1^2 + a_1^2 b_2^2 + a_2^1 b_1^1 + a_2^2 b_2^1 + a_2^1 b_1^2 + a_2^2 b_2^2 \\
\psi(V) &= a_1^1 b_1^1 + a_1^2 b_2^1 - a_1^1 b_1^2 - a_1^2 b_2^2 + a_2^1 b_1^1 + a_2^2 b_2^1 - a_2^1 b_1^2 - a_2^2 b_2^2 \\
\psi(H) &= a_1^1 b_1^1 + a_1^2 b_2^1 + a_1^1 b_1^2 + a_1^2 b_2^2 - a_2^1 b_1^1 - a_2^2 b_2^1 - a_2^1 b_1^2 - a_2^2 b_2^2 \\
\psi(D) &= a_1^1 b_1^1 + a_1^2 b_2^1 - a_1^1 b_1^2 - a_1^2 b_2^2 - a_2^1 b_1^1 - a_2^2 b_2^1 + a_2^1 b_1^2 + a_2^2 b_2^2
\end{aligned}
$$

$$
W(A) \cdot W(B) = \frac{1}{4} \begin{pmatrix} W_A & W_V \\ W_H & W_D \end{pmatrix}
$$

$$
\begin{aligned}
W_A &= (a_1^1 + a_2^1 + a_1^2 + a_2^2)(b_1^1 + b_2^1 + b_1^2 + b_2^2) + (a_1^1 + a_2^1 - a_1^2 - a_2^2)(b_1^1 - b_2^1 + b_1^2 - b_2^2) \\
W_V &= (a_1^1 + a_2^1 + a_1^2 + a_2^2)(b_1^1 + b_2^1 - b_1^2 - b_2^2) + (a_1^1 + a_2^1 - a_1^2 - a_2^2)(b_1^1 - b_2^1 - b_1^2 + b_2^2) \\
W_H &= (a_1^1 - a_2^1 + a_1^2 - a_2^2)(b_1^1 + b_2^1 + b_1^2 + b_2^2) + (a_1^1 - a_2^1 - a_1^2 + a_2^2)(b_1^1 - b_2^1 + b_1^2 - b_2^2) \\
W_D &= (a_1^1 - a_2^1 + a_1^2 - a_2^2)(b_1^1 + b_2^1 - b_1^2 - b_2^2) + (a_1^1 - a_2^1 - a_1^2 + a_2^2)(b_1^1 - b_2^1 - b_1^2 + b_2^2)
\end{aligned}
$$

$$
\begin{aligned}
W_A &= a_1^1b_1^1 + a_2^1b_1^1 + a_1^2b_2^1 + a_2^2b_2^1 + a_1^1b_1^2 + a_2^1b_1^2 + a_1^2b_2^2 + a_2^2b_2^2 \\
W_V &= a_1^1b_1^1 + a_2^1b_1^1 + a_1^2b_2^1 + a_2^2b_2^1 - a_1^1b_1^2 - a_2^1b_1^2 - a_1^2b_2^2 - a_2^2b_2^2 \\
W_H &= a_1^1b_1^1 - a_2^1b_1^1 + a_1^2b_2^1 - a_2^2b_2^1 + a_1^1b_1^2 - a_2^1b_1^2 + a_1^2b_2^2 - a_2^2b_2^2 \\
W_D &= a_1^1b_1^1 - a_2^1b_1^1 + a_1^2b_2^1 - a_2^2b_2^1 - a_1^1b_1^2 + a_2^1b_1^2 - a_1^2b_2^2 + a_2^2b_2^2
\end{aligned}
$$

This can then be compared to the coefficients of $W(A \cdot B)$ which were

$$
\begin{aligned}
\psi(A) &= a_1^1b_1^1 + a_1^2b_2^1 + a_1^1b_1^2 + a_1^2b_2^2 + a_2^1b_1^1 + a_2^2b_2^1 + a_2^1b_1^2 + a_2^2b_2^2 \\
\psi(V) &= a_1^1b_1^1 + a_1^2b_2^1 - a_1^1b_1^2 - a_1^2b_2^2 + a_2^1b_1^1 + a_2^2b_2^1 - a_2^1b_1^2 - a_2^2b_2^2 \\
\psi(H) &= a_1^1b_1^1 + a_1^2b_2^1 + a_1^1b_1^2 + a_1^2b_2^2 - a_2^1b_1^1 - a_2^2b_2^1 - a_2^1b_1^2 - a_2^2b_2^2 \\
\psi(D) &= a_1^1b_1^1 + a_1^2b_2^1 - a_1^1b_1^2 - a_1^2b_2^2 - a_2^1b_1^1 - a_2^2b_2^1 + a_2^1b_1^2 + a_2^2b_2^2
\end{aligned}
$$

Notice that $W(A) \cdot W(B) = W(A \cdot B)$, in the case of 2×2 matrices.

Qualifications for Wavelet Matrix Multiplication

1. Are the matrices the same size and are they each a square matrix?
2. If not, are the dimensions suitable for multiplication?
3. If so, what is the maximum resolution for each matrix? The lesser maximum is the limit for both.
4. Is the wavelet transform performed on each matrix the same?

Wavelet Based Matrix Multiply Procedure

The general wavelet based matrix multiply is as follows:

1. Arguments:
   - A : a $m \times p$ matrix
   - B: a $p \times n$ matrix
2. Results: C : a $m \times n$ matrix
3. Procedure:
   - $A \xrightarrow{\psi} \alpha$
   - $B \xrightarrow{\psi} \beta$
   - $\alpha \stackrel{ChainRow}{\longrightarrow} \alpha^c$
   - $\beta \stackrel{ChainColumn}{\longrightarrow} \beta^c$
   - Chain Multiply $(\alpha^c, \beta^c) \to C$

Partial Differential Equations

- Overview of PDE
- Related Work
- Proposed Work
- Partial Differential Equations (PDE) in general
- Classic Methods
- PDE problems
- Related Wavelet Research

PDE In General

General PDE Equation

$$Au_{xx} + Bu_{xy} + Cu_{yy} + Du_x + Eu_y + Fu = G$$

- The condition of the famous equation $B^2 - 4AC$ determines the category of PDE:
  - Less than zero is an elliptical PDE
  - More than zero is a hyperbolic PDE
  - Equal to zero is parabolic PDE
- Solution to PDE numerically includes explicit, implicit, Galerkin, and Monte Carlo

Elliptical Partial Differential Equation

- In general, elliptical PDE measure the strength of an item compared to its neighbors.
- Physical Examples include:
  - Heat Equations $u_t = \alpha^2 \nabla^2 u$
  - Wave Equations $u_{tt} = \alpha^2 \nabla^2 u$
  - Laplace Equation $\nabla^2 u = 0$
  - Poison's Formula $\frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2} = 0$
- Boundary Conditions for setting up a PDE solution:
  - Dirichlet
  - Neumann
  - Robbin's

Parabolic Partial Differential Equations

- Parabolic PDE model diffusion and fluid mechanics
- Common uses to today
  - Gas dispersion in a room
  - Fluid dispersion within a pipe or river

Hyperbolic Partial Differential Equations

- Models oscillating phenomena (EM Fields, Vibrating Strings)
- Analytic Solution D'Alembert Solution
  - Replace (x,t) by the new canonical coordinates
  - Solve the transformed equations
  - Transformed equations
  - Transform the solution into original coordinates
  - Substitute the general solution into the IC's to acquire the constants
- Beware of Boundary Conditions: Examples (fixed or variable)

$$u_n + \lambda u = g(t)$$

$$u = g(t)$$

$$u_n = g(t)$$

Difference Equations

- Central Difference Formula
  - $f'(x) \approx \frac{f(x+h) - f(x-h)}{2h}$
  - $f''(x) \approx \frac{f(x+h) - 2f(x) + f(x-h)}{h^2}$
- Backward Difference Formula
  - $f'(x) \approx \frac{f(x) - f(x-h)}{h}$
- Forward Difference Formula
  - $f'(x) \approx \frac{f(x+h) - f(x)}{h}$

Explicit Method

- Basic Method:
    - Start with the index value of 0 (i=0 at the initial value)
    - Find the solution $\forall x \in u(x,t), t = t_{i+1}$
    - Establish the boundary conditions with respect for $u_{t_{i+1},x}$ by boundary condition approximation formula.
    - Repeat the previous two steps until i = n (where n is the largest index)
- Strength: It is a top-down dynamic algorithm
- Weakness: Numerically, it is unstable. Even though it is sound by Turing Theory

Implicit Method

- Pick some value for λ such that $\lambda \in [0, 1]$
- Pick Δx and Δt and assign grid points
- Use computational molecule to generate equation
- Solve Matrix

PDE Problems

- Semi-Infinite String Problem
- Heat Diffusion
- Diffusion - Convection

Semi Infinite String Problem

- The semi-infinite string problem is a typical resonance problem.
  - PDE $u_{tt} = c^2 u_{xx} \quad \forall x \in (0,\infty)$ and $\forall t \in (0,\infty)$
  - BC $u(0,t) = 0$
  - IC
    - $u(0,x) = f(x)$
    - $u_t(x,0) = g(x)$
  - general solution: $u(x,t) = \frac{1}{2}[f(x-ct) + f(x+ct)] + \frac{1}{2c}\int^{x+ct} x - ct g(\zeta)d\zeta$
  - $c^2 u_{xx}$ is the net force due to the tension on the string.
  - u_{tt} represents the longitudinal or torsional vibrations on the string.

Heat Diffusion

- The heat diffusion/ conduction equation defines heat in a solid at any point and time within the domain.
- Diffusitivity variable defined: $\alpha = \frac{K}{\tau\sigma}$
  - K is the thermal constant
  - τ is the density
  - σ is the specific heat
- PDE Defined:
  - $u_t = \alpha^2 u_{xx}$ such that
  - u_t is the change in temperature with respect to time.
  - u_{xx} is the concavity of the temperature

Diffusion - Convection

- The big idea for diffusion convection is that convection currents effects on the diffusion process.
- Generic Formula: $u_t = \alpha^2 u_{xx} - v u_x$ such that
  - u_t is the change in time
  - $\alpha^2 u_{xx}$ is the diffusion component
  - $v u_x$ is the convection component
  - v is the velocity of the convection current
  - α is the diffusitivity constant

Proposed Work on PDE

- Apply wavelet operator to partial differential equations
- Determine levels of accuracy and approximation
- Determine efficiency