Parallel Programming Notes

Ayush Raina

March 6, 2025

Sequential Fast Fourier Transform

Given an input array $X[1:n]$, we need to output the transformed array $Y[1:n]$ where $Y[i] = \sum_{k=0}^{n-1} X[k]\omega^{ik}$ and $0 \leq i \leq n$ where ω^{ik} is called the twiddle factor and is the primitive n^{th} root of unity. $\omega = e^{-2\pi\sqrt{-1}/n}$ and $i = \sqrt{-1}$ generally called as iota.

We can further simplify the computation as follows: $Y[i] = \sum_{k=0}^{\frac{n}{2}-1} X[2k]\omega^{2ik} + \sum_{k=0}^{\frac{n}{2}-1} X[2k+1]\omega^{(2k+1)i}$
$Y[i] = \sum_{k=0}^{\frac{n}{2}-1} X[2k]\omega^{2ik} + \omega^{i} \sum_{k=0}^{\frac{n}{2}-1} X[2k+1]\omega^{2ik}$

Putting the value of $\omega = e^{-2\pi i/n}$, we get the final Expression as:
Add the remaining piece here

Sequential FFT - Recursive Solution

```

procedure R_FFT(X,Y,n,w)
if(n == 1) then Y[0] = X[0]

begin
  Add this
```

Sequential FFT - Iterative Solution

```
  procedure I_FFT(X,Y,n)
```