Butterfly as Linear Algebra

Danny Hermes

February 4, 2015

Table of contents

Extract Source of PDF

NOTE: To extract the source from this PDF, execute:

```
pdftk butterfly_slides.pdf unpack_files output .
```

Motivation

We want to compute

$$\widehat{f}_k = \sum_{j=0}^{N-1} e^{t_k s_j} f_j$$

for $0 \leq k < N$ and $t_k, s_j \in \mathbf{R}$.

Kernel Approximation

For $z \in \mathbf{C}$:

$$e^z = \sum_{\alpha=0}^{m} \frac{z^\alpha}{\alpha!} + E_m$$

with error $|E_m| \leq \left(\frac{|z|e}{m}\right)^m$.

Basic Trick

We can show that

$$ts = \tau\sigma + \tau(s - \sigma) + \sigma(t - \tau) + (t - \tau)(s - \sigma)$$

for any σ, τ.

Basic Trick in Use

We can use it to say

$$
\begin{aligned}
e^{ts} f &= e^{\tau\sigma} e^{\tau(s-\sigma)} e^{\sigma(t-\tau)} e^{(t-\tau)(s-\sigma)} f \\
&= e^{\tau\sigma} e^{\tau(s-\sigma)} e^{\sigma(t-\tau)} \sum_{\alpha \geq 0} \frac{1}{\alpha!} (t-\tau)^{\alpha} (s-\sigma)^{\alpha} f \\
&= \sum_{\alpha \geq 0} \left\{ \left[\frac{1}{\alpha!} e^{\tau\sigma} e^{\tau(s-\sigma)} (s-\sigma)^{\alpha} \right] e^{\sigma(t-\tau)} (t-\tau)^{\alpha} f \right\} \\
&= \sum_{\alpha \geq 0} K(s; \alpha, \tau, \sigma) e^{\sigma(t-\tau)} (t-\tau)^{\alpha} f.
\end{aligned}
$$

Basic Trick Emphasis

We emphasize the definition:

$$\boxed{K(s;\alpha,\tau,\sigma) = \frac{1}{\alpha!} e^{\tau\sigma} e^{\tau(s-\sigma)} (s-\sigma)^{\alpha}}.$$

Partitioning the Inputs

For a given level ℓ, assume we have partitioned the target space T into 2^ℓ evenly spaced intervals and have partitioned S into $2^{L-\ell}$ where $L = \mathcal{O}(\log_2 N)$ is some maximum depth which we determine later.

For example if $T \subset [0, 1]$ then

$$\begin{aligned}
T\{1\} &= \left[0, \frac{1}{2}\right] && \cup \left[\frac{1}{2}, 1\right] \\
T\{2\} &= \left[0, \frac{1}{4}\right] \cup \left[\frac{1}{4}, \frac{1}{2}\right] && \cup \left[\frac{1}{2}, \frac{3}{4}\right] \cup \left[\frac{3}{4}, 1\right].
\end{aligned}$$

Partitioning the Inputs

In general

$$T\{\ell\} = \bigcup_{i=0}^{2^{\ell}-1} T(i;\ell)$$

and

$$S\{L-\ell\} = \bigcup_{j=0}^{2^{L-\ell}-1} S(j;L-\ell)\,.$$

Define the centers of these boxes (intervals here, but feel free to get more generic) as

$$\boxed{\tau(i;\ell) \ \text{ and } \ \sigma(m;L-\ell)}.$$

Return to Motivation

Recall

$$\widehat{f}_k = \sum_{j=0}^{N-1} e^{t_k s_j} f_j$$

and notice that for a fixed k and fixed ℓ, there is a unique i such that $t_k \in T(i;\ell)$, then invoking $K(s;\alpha,\tau(i;\ell),\sigma)$ as σ varies over

$$\sigma(0;L-\ell),\ldots,\sigma\left(2^{L-\ell}-1;L-\ell\right)$$

will turn our sum into $2^{L-\ell}$ sums (each with another sum over α).

Return to Motivation

Thus, letting $\tau = \tau(i;\ell)$:

$$\begin{aligned}\widehat{f}_k &= \sum_{j=0}^{N-1} e^{t_k s_j} f_j \\ &= \sum_m \sum_{s \in S(m;L-\ell)} e^{t_k s} f(s) \\ &= \sum_m \sum_{s \in S(m)} \sum_{\alpha \geq 0} K\left(s;\alpha,\tau,\sigma(m)\right) e^{\sigma(m)(t_k-\tau)} (t_k-\tau)^\alpha f(s).\end{aligned}$$

Above we write $\sigma(m)$ instead of $\sigma(m;L-\ell)$ since the refinement level is clear from context.

NOTE: We write $f(s)$ to denote f_j in the case that $s = s_j$.

Rearrangement

Rearranging

$$\begin{aligned}
\widehat{f}_k &= \sum_m \sum_{s\in S(m)} \sum_{\alpha\geq 0} K\left(s;\alpha,\tau,\sigma(m)\right) e^{\sigma(m)(t_k-\tau)}(t_k-\tau)^\alpha f(s) \\
&= \sum_m \sum_{\alpha\geq 0} e^{\sigma(m)(t_k-\tau)}(t_k-\tau)^\alpha \sum_{s\in S(m)} K\left(s;\alpha,\tau,\sigma(m)\right) f(s) \\
&= \sum_m \sum_{\alpha\geq 0} C\left(\alpha,\ell,i,m\right) e^{\sigma(m)(t_k-\tau)}(t_k-\tau)^\alpha .
\end{aligned}$$

Rearrangement

We emphasize the definition:

$$C(\alpha, \ell, i, m) = \sum_{s \in S(m)} K(s; \alpha, \tau(i), \sigma(m)) f(s)$$

or more explicitly

$$\boxed{C(\alpha, \ell, i, m) = \frac{1}{\alpha!} \sum_{s \in S(m)} e^{\tau(i)\sigma(m)} e^{\tau(i)(s-\sigma(m))} (s - \sigma(m))^{\alpha} f(s)}.$$

Approximation

For any level ℓ with $t_k \in T(i;\ell)$ we can approximate the exact value

$$\widehat{f}_k = \sum_m \sum_{\alpha \geq 0} C\left(\alpha, \ell, i, m\right) e^{\sigma(m)(t_k - \tau(i))}(t_k - \tau(i))^{\alpha}.$$

by

$$\widehat{f}_k \approx \sum_m \sum_{\alpha = 0}^{M-1} C\left(\alpha, \ell, i, m\right) e^{\sigma(m)(t_k - \tau(i))}(t_k - \tau(i))^{\alpha}$$

for some max number of terms M.

Approximation

Notice that $C(\alpha, \ell, i, m)$ does not depend on k, so we only need to compute this for M choices of α, 2^{ℓ} choices of i and $2^{L-\ell}$ choices of m. In total

$$M \cdot 2^{L} = \mathcal{O}(M \cdot N) = \mathcal{O}(N)$$

independent of ℓ.

Trying to Compute These

We start with $\ell = 0$ and consider what it might take to compute.

Trying to Compute These

We start with $\ell = 0$ and consider what it might take to compute.

- In this case we have exactly one target box $T(0;0)$ and $2^L = \mathcal{O}(N)$ source boxes $S(m;L)$.

Trying to Compute These

We start with $\ell = 0$ and consider what it might take to compute.

- In this case we have exactly one target box $T(0;0)$ and $2^L = \mathcal{O}(N)$ source boxes $S(m;L)$.
- Since at level L, we know $|S(m;L)| = \mathcal{O}(1)$ hence computing each coefficient is $\mathcal{O}(1)$.

Trying to Compute These

We start with $\ell = 0$ and consider what it might take to compute.

- In this case we have exactly one target box $T(0;0)$ and $2^L = \mathcal{O}(N)$ source boxes $S(m;L)$.
- Since at level L, we know $|S(m;L)| = \mathcal{O}(1)$ hence computing each coefficient is $\mathcal{O}(1)$.
- However, for each of these boxes, we need to compute $\exp\{\sigma(m;L)(t_k - \tau(0;0))\}$ which is $\mathcal{O}\left(N^2\right)$ since we have $2^L = \mathcal{O}(N)$ choices for m and $N = \mathcal{O}(N)$ choices for t_k.

Trying to Compute These

Instead we consider the **end** where $\ell = L$.

Trying to Compute These

Instead we consider the **end** where $\ell = L$.

- In this case we have exactly one source box $S(0; 0)$ and $2^L = \mathcal{O}(N)$ target boxes $T(i; L)$.

Trying to Compute These

Instead we consider the **end** where $\ell = L$.

- In this case we have exactly one source box $S(0;0)$ and $2^L = \mathcal{O}(N)$ target boxes $T(i;L)$.
- Since at level 0, we know $|S(0;0)| = N = \mathcal{O}(N)$ hence computing each coefficient (for fixed i) is $\mathcal{O}(N)$.

Trying to Compute These

Instead we consider the **end** where $\ell = L$.

- In this case we have exactly one source box $S(0;0)$ and $2^L = \mathcal{O}(N)$ target boxes $T(i;L)$.
- Since at level 0, we know $|S(0;0)| = N = \mathcal{O}(N)$ hence computing each coefficient (for fixed i) is $\mathcal{O}(N)$.
- Since we have $\mathcal{O}(N)$ choices for i, there are $\mathcal{O}\left(N^2\right)$ coefficients to compute.

Here's the Catch

The the $\ell = L$ approach sinks us immediately by asking us to compute $\mathcal{O}\left(N^2\right)$ coefficients, we have a **positive takeaway**.

Since there is only one value $\sigma = \sigma(0; 0)$, we can compute each value as

$$\widehat{f_k} \approx \sum_{\alpha=0}^{M-1} C\left(\alpha, L, i, m = 0\right) e^{\sigma\left(t_k - \tau(i;L)\right)} (t_k - \tau(i; L))^{\alpha}.$$

$$\boxed{\text{This is happily } \mathcal{O}(M) = \mathcal{O}(1) \text{ to compute.}}$$

Now What

- We've seen that no matter what ℓ is, we have $M \cdot 2^L = \mathcal{O}(N)$ coefficients $C(\alpha, \ell, i, m)$ to compute.

Now What

- We've seen that no matter what ℓ is, we have $M \cdot 2^L = \mathcal{O}(N)$ coefficients $C(\alpha, \ell, i, m)$ to compute.
- When $\ell = 0$ we can compute all the coefficients in $\mathcal{O}(1)$ since the refinement $L - \ell = L$ in the source space gives $|S(m; L)| = \mathcal{O}(1)$.

Now What

- We've seen that no matter what ℓ is, we have $M \cdot 2^L = \mathcal{O}(N)$ coefficients $C(\alpha, \ell, i, m)$ to compute.
- When $\ell = 0$ we can compute all the coefficients in $\mathcal{O}(1)$ since the refinement $L - \ell = L$ in the source space gives $|S(m; L)| = \mathcal{O}(1)$.
- When $\ell = L$ we can compute all the values $\widehat{f}_k$ in $\mathcal{O}(1)$ provided we already have the $C(\alpha, \ell, i, m)$.

Now What

- We've seen that no matter what ℓ is, we have $M \cdot 2^L = \mathcal{O}(N)$ coefficients $C(\alpha, \ell, i, m)$ to compute.
- When $\ell = 0$ we can compute all the coefficients in $\mathcal{O}(1)$ since the refinement $L - \ell = L$ in the source space gives $|S(m; L)| = \mathcal{O}(1)$.
- When $\ell = L$ we can compute all the values $\widehat{f}_k$ in $\mathcal{O}(1)$ provided we already have the $C(\alpha, \ell, i, m)$.
- This means that the computation of $\left\{\widehat{f}_k\right\}_k$ is $\mathcal{O}(N)$.

Now What

- To summarize, we know we can compute the $\mathcal{O}(N)$ values $C(\alpha, 0, i = 0, m)$ in $\mathcal{O}(N)$ and we know we can use the $\mathcal{O}(N)$ values $C(\alpha, L, i, m = 0)$ to compute $\left\{\widehat{f}_k\right\}_k$ in $\mathcal{O}(N)$.

Now What

- To summarize, we know we can compute the $\mathcal{O}(N)$ values $C(\alpha, 0, i = 0, m)$ in $\mathcal{O}(N)$ and we know we can use the $\mathcal{O}(N)$ values $C(\alpha, L, i, m = 0)$ to compute $\left\{\widehat{f}_k\right\}_k$ in $\mathcal{O}(N)$.
- (Let's assume that) Butterfly gives a process to convert

$$\{C(\alpha, 0, i, m)\} \to \{C(\alpha, 1, i, m)\} \to \cdots \to \{C(\alpha, L, i, m)\}$$

using $\mathcal{O}(N \log N)$ operations.

Now What

- To summarize, we know we can compute the $\mathcal{O}(N)$ values $C(\alpha, 0, i = 0, m)$ in $\mathcal{O}(N)$ and we know we can use the $\mathcal{O}(N)$ values $C(\alpha, L, i, m = 0)$ to compute $\left\{\widehat{f}_k\right\}_k$ in $\mathcal{O}(N)$.
- (Let's assume that) Butterfly gives a process to convert

$$\{C(\alpha, 0, i, m)\} \to \{C(\alpha, 1, i, m)\} \to \cdots \to \{C(\alpha, L, i, m)\}$$

using $\mathcal{O}(N \log N)$ operations.
- Thus, we can compute $\left\{\widehat{f}_k\right\}_k$ in

$$\mathcal{O}(N) + \mathcal{O}(N \log N) + \mathcal{O}(N) = \mathcal{O}(N \log N).$$

Now What

- To summarize, we know we can compute the $\mathcal{O}(N)$ values $C(\alpha, 0, i = 0, m)$ in $\mathcal{O}(N)$ and we know we can use the $\mathcal{O}(N)$ values $C(\alpha, L, i, m = 0)$ to compute $\left\{\widehat{f}_k\right\}_k$ in $\mathcal{O}(N)$.
- (Let's assume that) Butterfly gives a process to convert

$$\{C(\alpha, 0, i, m)\} \to \{C(\alpha, 1, i, m)\} \to \cdots \to \{C(\alpha, L, i, m)\}$$

  using $\mathcal{O}(N \log N)$ operations.
- Thus, we can compute $\left\{\widehat{f}_k\right\}_k$ in

$$\mathcal{O}(N) + \mathcal{O}(N \log N) + \mathcal{O}(N) = \mathcal{O}(N \log N).$$

- **In other words, the process of converting the coefficients will dominate**.

High Level Approach

In order to facilitate the process

$$\{C(\alpha, \ell, i, m)\}_{\alpha,i,m} \rightarrow \{C(\alpha, \ell+1, i, m)\}_{\alpha,i,m}$$

we split into two parts:

- Refine T and compute interaction coefficients for $T\{\ell+1\}$ and $S\{L-\ell\}$.
- This first step will use the existing interaction coefficients for $T\{\ell\}$ and $S\{L-\ell\}$. These are exactly $\{C(\alpha, \ell, i, m)\}_{\alpha,i,m}$.
- Coarsen S and compute interaction coefficients for $T\{\ell+1\}$ and $S\{L-\ell-1\}$ using the intermediate coefficients computed in the first step. This will produce $\{C(\alpha, \ell+1, i, m)\}_{\alpha,i,m}$.

Memories...

Before we get started, recall for $\tau = \tau(i; \ell)$ and $\sigma = \sigma(m; L - \ell)$:

$$C(\alpha, \ell, i, m) = \frac{1}{\alpha!} \sum_{s \in S(m; L-\ell)} e^{\tau\sigma} e^{\tau(s-\sigma)} (s - \sigma)^{\alpha} f(s).$$

Refining T

When refining $T(i;\ell)$, we get two new intervals. Assuming they are ordered by their indices, we have

$$T(i;\ell) = T(2i;\ell+1) \cup T(2i+1;\ell+1).$$

For shorthand, we'll use the notation

$$\tau(i) = \tau(i;\ell) \text{ and } \tau^+(i') = \tau(i';\ell+1)$$

when the value of ℓ is clear from context.

In addition, we define the intermediate interaction coefficients:

$$\boxed{C^+(\alpha,\ell,i',m) = \frac{1}{\alpha!}\sum_{s\in S(m)} e^{\tau^+(i')\sigma(m)} e^{\tau^+(i')(s-\sigma(m))}(s-\sigma(m))^\alpha f(s)}.$$

Refining T

One can show

$$\boxed{C^{+}(\alpha, \ell, i', m) \approx e^{(\tau^{+}-\tau)\sigma} \sum_{\beta=0}^{M-1} (\tau^{+}-\tau)^{\beta} \binom{\alpha+\beta}{\beta} C(\alpha+\beta, \ell, i, m)}$$

by truncating the sum

$$e^{(\tau^{+}-\tau)(s-\sigma)} = \sum_{\beta \geq 0} \frac{1}{\beta!} (\tau^{+}-\tau)^{\beta} (s-\sigma)^{\beta}$$

and rewriting

$$e^{\tau^{+}\sigma} e^{\tau^{+}(s-\sigma)} = e^{(\tau^{+}-\tau)\sigma} e^{\tau\sigma} e^{\tau(s-\sigma)} e^{(\tau^{+}-\tau)(s-\sigma)}.$$

Refining T

In reality, $C(\alpha + \beta, \ell, i, m)$ isn't defined for all values of β since we only consider inputs to $C(\,\cdot\,, \ell, i, m)$ from $[0, M)$.

Thus

$$\boxed{C^+(\alpha, \ell, i', m) \approx e^{(\tau^+ - \tau)\sigma} \sum_{\gamma=\alpha}^{M-1} (\tau^+ - \tau)^{\gamma - \alpha} \binom{\gamma}{\alpha} C(\gamma, \ell, i, m)}$$

is the actual approximation we use.

Coarsening S

Now we have written the $C^+(\,\cdot\,, \ell, i', m)$ values in terms of the $C(\,\cdot\,, \ell, i, m)$ values.

From here, we want to use the $C^+(\,\cdot\,, \ell, i', m)$ values to express the $C(\,\cdot\,, \ell+1, i', m')$ values in the same fashion, but here by coarsening S:

$$S(m'; L-\ell-1) = S(2m'; L-\ell) \cup S(2m'+1; L-\ell).$$

The definition m' here requires that $m \in \{2m', 2m'+1\}$. In either case $m' = \left\lfloor \frac{m}{2} \right\rfloor$.

Similarly, for shorthand, we'll use the notation

$$\sigma(m) = \sigma(m; L-\ell) \text{ and } \sigma^-(m') = \sigma(m'; L-\ell-1).$$

Coarsening S

When computing

$$C(\alpha, \ell+1, i', m') = \frac{1}{\alpha!} \sum_{s \in S(m';L-\ell-1)} e^{\tau^+ \sigma^-} e^{\tau^+(s-\sigma^-)} (s-\sigma^-)^\alpha f(s)$$

we need to split our sum into

$$\sum_{s \in S(m';L-\ell-1)} = \sum_{s \in S(2m';L-\ell)} + \sum_{s \in S(2m'+1;L-\ell)} .$$

Coarsening S

After splitting the into two sums, we can show

$$C(\alpha, \ell+1, i', m') = \sum_{\beta=0}^{\alpha} \frac{(\sigma(2m') - \sigma^-)^{\alpha-\beta}}{(\alpha-\beta)!} C^+\left(\beta, \ell, i', 2m'\right) + \sum_{\beta=0}^{\alpha} \frac{(\sigma(2m'+1) - \sigma^-)^{\alpha-\beta}}{(\alpha-\beta)!} C^+\left(\beta, \ell, i', 2m'+1\right).$$

by using binomial expansion

$$(s-\sigma^-)^{\alpha} = \sum_{\beta=0}^{\alpha} \binom{\alpha}{\beta} (s-\sigma)^{\beta} (\sigma-\sigma^-)^{\alpha-\beta}.$$

Combining Transformations

For each fixed ℓ, i, m, the coefficient vector $C(\,\cdot\,, \ell, i, m) \in \mathbf{R}^M$ and the system we've described by refining and coarsening gives a block linear transformation:

$$A : \left[\begin{array}{c} C(\,\cdot\,, \ell, i, 2m') \\ C(\,\cdot\,, \ell, i, 2m'+1) \end{array} \right] \mapsto \left[\begin{array}{c} C(\,\cdot\,, \ell+1, 2i, m') \\ C(\,\cdot\,, \ell+1, 2i+1, m') \end{array} \right].$$

In the case of higher dimensions or for inputs in $\mathbf{C}$ the transformation goes from operating on vectors of size $2M$ to size $2^d M$ where d is the dimension (over $\mathbf{R}$) of the space containing the t_k, s_j.

Determine Matrix A

To determine each element A_{pq} we need to find the coefficient of the q^{th} element of the input vector in the p^{th} element of the output vector.

This splits into four distinct cases, depending on whether each of p, q correspond to the top or bottom half. Thus we can split A into a block 2×2 matrix

$$A = \left[\begin{array}{cc} E & F \\ G & H \end{array} \right].$$

Determine Matrix A

For $0 \leq p, q < M$.

$$\begin{aligned}
C(p, \ell+1, 2i, m') &= \cdots + E_{pq} C(q, \ell, i, 2m') + \cdots \\
C(p, \ell+1, 2i, m') &= \cdots + F_{pq} C(q, \ell, i, 2m'+1) + \cdots \\
C(p, \ell+1, 2i+1, m') &= \cdots + G_{pq} C(q, \ell, i, 2m') + \cdots \\
C(p, \ell+1, 2i+1, m') &= \cdots + H_{pq} C(q, \ell, i, 2m'+1) + \cdots
\end{aligned}$$

Determine General Block Submatrix X

With $i' \in \{2i, 2i+1\}$ and $m \in \{2m', 2m'+1\}$

$$C(p, \ell+1, i', m') = \cdots + X_{pq} C(q, \ell, i, m) + \cdots$$

we use $\sigma = \sigma(m)$, $\sigma^- = \sigma(m')$,

$$C(p, \ell+1, i', m') = \cdots + \sum_{\beta=0}^{p} \frac{(\sigma - \sigma^-)^{p-\beta}}{(p-\beta)!} C^+(\beta, \ell, i', m) + \cdots$$

and recall with $\tau = \tau(i)$, $\tau^+ = \tau(i')$,

$$C^+(\beta, \ell, i', m) = e^{(\tau^+ - \tau)\sigma} \sum_{\gamma=\beta}^{M-1} (\tau^+ - \tau)^{\gamma-\beta} \binom{\gamma}{\beta} C(\gamma, \ell, i, m).$$

Determine General Block Submatrix X

Since

$$\sum_{\beta=0}^{p}\sum_{\gamma=\beta}^{M-1} = \sum_{\gamma=0}^{M-1}\sum_{\beta=0}^{\min(p,\gamma)},$$

the coefficient of $C(\gamma,\cdots)$ when $\gamma = q$ is

$$X_{pq} = e^{(\tau^{+}-\tau)\sigma}\sum_{\beta=0}^{\min(p,q)}\frac{(\sigma-\sigma^{-})^{p-\beta}}{(p-\beta)!}(\tau^{+}-\tau)^{q-\beta}\binom{q}{\beta}.$$

Explicit Blocks: $m = 2m'$

- In both cases $\sigma = \sigma(2m'), \sigma^- = \sigma(m')$ and $\tau = \tau(i)$.
- When $\tau^+ = \tau(2i)$,

$$E_{pq} = e^{(\tau^+ - \tau)\sigma} \sum_{\beta=0}^{\min(p,q)} \frac{(\sigma - \sigma^-)^{p-\beta}}{(p-\beta)!} (\tau^+ - \tau)^{q-\beta} \binom{q}{\beta}.$$

- When $\tau^+ = \tau(2i+1)$,

$$G_{pq} = e^{(\tau^+ - \tau)\sigma} \sum_{\beta=0}^{\min(p,q)} \frac{(\sigma - \sigma^-)^{p-\beta}}{(p-\beta)!} (\tau^+ - \tau)^{q-\beta} \binom{q}{\beta}.$$

Explicit Blocks: $m = 2m' + 1$

- In both cases $\sigma = \sigma(2m'+1), \sigma^- = \sigma(m')$ and $\tau = \tau(i)$.
- When $\tau^+ = \tau(2i)$,

$$F_{pq} = e^{(\tau^+ - \tau)\sigma} \sum_{\beta=0}^{\min(p,q)} \frac{(\sigma - \sigma^-)^{p-\beta}}{(p-\beta)!} (\tau^+ - \tau)^{q-\beta} \binom{q}{\beta}.$$

- When $\tau^+ = \tau(2i+1)$,

$$H_{pq} = e^{(\tau^+ - \tau)\sigma} \sum_{\beta=0}^{\min(p,q)} \frac{(\sigma - \sigma^-)^{p-\beta}}{(p-\beta)!} (\tau^+ - \tau)^{q-\beta} \binom{q}{\beta}.$$

Potential Optimization

We encounter

$$\frac{(\sigma - \sigma^-)^{p-\beta}}{(p-\beta)!}(\tau^+ - \tau)^{q-\beta}\binom{q}{\beta}$$

for various values of $\sigma, \sigma^-, \tau, \tau^+$ but for a fixed set of choices of p, q (throughout the life of the code).

We can write this instead as

$$\frac{q!}{\beta!}\frac{(\sigma - \sigma^-)^{p-\beta}}{(p-\beta)!}\frac{(\tau^+ - \tau)^{q-\beta}}{(q-\beta)!}.$$

Operation Count

- Constructing each matrix A at the $\ell \to \ell + 1$ step (for fixed i, m') requires creating $(2M)^2$ matrix entries.

Operation Count

- Constructing each matrix A at the $\ell \to \ell + 1$ step (for fixed i, m') requires creating $(2M)^2$ matrix entries.
- Each p, q entry in each of the 2×2 block submatrices can be constructed in $\min(p, q)$ hence can be computed in $\mathcal{O}(M)$.

Operation Count

- Constructing each matrix A at the $\ell \to \ell + 1$ step (for fixed i, m') requires creating $(2M)^2$ matrix entries.
- Each p, q entry in each of the 2×2 block submatrices can be constructed in $\min(p, q)$ hence can be computed in $\mathcal{O}(M)$.
- Each A is applied to 2 of the 2^L coefficient sets and this applicated requires $\mathcal{O}\left((2M)^3\right)$ operations.

Operation Count

- Constructing each matrix A at the $\ell \to \ell + 1$ step (for fixed i, m') requires creating $(2M)^2$ matrix entries.
- Each p, q entry in each of the 2×2 block submatrices can be constructed in $\min(p, q)$ hence can be computed in $\mathcal{O}(M)$.
- Each A is applied to 2 of the 2^L coefficient sets and this applicated requires $\mathcal{O}\left((2M)^3\right)$ operations.
- Thus the total work to convert two of the coefficient sets is $(2M)^2\mathcal{O}(M) + \mathcal{O}\left((2M)^3\right) = \mathcal{O}\left(M^3\right)$.

Operation Count

- Since we have $2^L/2 = 2^{L-1} = \mathcal{O}(N/2)$ such pairs of coefficient sets, the total work is $\mathcal{O}\left(M^3 \cdot N\right)$.

Operation Count

- Since we have $2^L/2 = 2^{L-1} = \mathcal{O}(N/2)$ such pairs of coefficient sets, the total work is $\mathcal{O}\left(M^3 \cdot N\right)$.
- When M — the max number of terms in the Taylor approximation of the kernel e^z — is reasonable, the work in

$$\{C(\alpha, \ell, i, m)\}_{\alpha,i,m} \to \{C(\alpha, \ell+1, i, m)\}_{\alpha,i,m}$$

is $\mathcal{O}(N)$.

Operation Count

- Since we have $2^L/2 = 2^{L-1} = \mathcal{O}(N/2)$ such pairs of coefficient sets, the total work is $\mathcal{O}\left(M^3 \cdot N\right)$.
- When M — the max number of terms in the Taylor approximation of the kernel e^z — is reasonable, the work in

$$\{C(\alpha, \ell, i, m)\}_{\alpha,i,m} \to \{C(\alpha, \ell+1, i, m)\}_{\alpha,i,m}$$

  is $\mathcal{O}(N)$.
- Since we only have L steps

$$\ell = 0 \to \ell = 1 \to \cdots \to \ell = L-1 \to \ell = L$$

  the total work is $\mathcal{O}(L \cdot N)$. As we already mentioned $L = \mathcal{O}(\log_2 N)$ hence the total work is $\boxed{\mathcal{O}(N \log N)}$.

Error Analysis

We have truncated the Taylor series for the kernel in two primary places:

$$\widehat{f}_k \approx \sum_m \sum_{\alpha=0}^{M-1} C\left(\alpha, \ell, i, m\right) e^{\sigma(m)(t_k - \tau(i))} (t_k - \tau(i))^{\alpha}$$

and

$$C^{+}(\alpha, \ell, i', m) \approx e^{(\tau^{+} - \tau)\sigma} \sum_{\gamma=\alpha}^{M-1} (\tau^{+} - \tau)^{\gamma-\alpha} \binom{\gamma}{\alpha} C(\gamma, \ell, i, m).$$

We need to understand how these errors propagate through our solution.