Barycentric Lagrange

Let's work on evaluating the Lagrange interpolator with knots in the xy-plane. Here we have $n+1$ nodes: $x_0 < x_1 < \ldots < x_n$, and $n+1$ knots: $(x_i, y_i),\ i = 0 : n$. The vector of y-values may just be data, or we may know that $y_i = f(x_i)$ for some function f. The Lagrange interpolator of degree n (or less) for a function f on these knots gives

$$f = P + R, \qquad \text{where} \qquad P = \sum_{i=0}^{n} y_j L_{nj}, \qquad R \text{ is the remainder,}$$

and we want to find the number $P(x)$, for some $x \in \mathbb{R}$ (maybe many such values).

Each Lagrange basis function, L_{nj}, is a polynomial of degree n, and we have $n+1$ of them to evaluate, so it looks like about $2n^2$ flops are required. We'll rewrite P in a way that cuts this to about $n^2/2$ flops. We begin by noting that the denominator of L_{nj} is a scalar which only depends on the nodes:

$$L_{nj}(x) = \prod_{i=0, i\neq j}^{n} \frac{x - x_i}{x_j - x_i} \quad \equiv \quad w_j \prod_{i=0, i\neq j}^{n} (x - x_i), \qquad \text{with} \qquad w_j = \frac{1}{\prod_{i=0, i\neq j}^{n} (x_j - x_i)}.$$

If we let $l(x) = \prod_{i=0}^{n} (x - x_i)$ (which doesn't depend on f) , then

$$L_{nj}(x) = w_j \frac{l(x)}{x - x_j} \qquad \text{and thus} \qquad P(x) = \sum_{i=0}^{n} \frac{y_j w_j l(x)}{x - x_j}.$$

Noticing that $l(x)$ is common to all terms, we can write

$$P(x) = l(x) \sum_{i=0}^{n} \frac{y_j w_j}{x - x_j}.$$

While this is a good starting point for computing $P(x)$, we can push it a bit further by using the Lagrange interpolator on these nodes for the function $g(x) \equiv 1$. The y_j for this g are $y_j = 1,\ j = 0 : n$, and since g is a polynomial of degree $\leq n$, its interpolator is g itself (there is no remainder). This gives the identity

$$l(x) \sum_{i=0}^{n} \frac{w_j}{x - x_j} = 1 \qquad \text{for all } x.$$

Dividing $P(x)$ by 1 (and cancelling the $l(x)$ factors) gives the *Barycentric form* of the Lagrange interpolator:

$$P(x) = \frac{\displaystyle\sum_{i=0}^{n} \frac{y_j w_j}{x - x_j}}{\displaystyle\sum_{i=0}^{n} \frac{w_j}{x - x_j}}.$$

This form requires computing the w_j's (which only depend on the knots, but be careful of over/under-flow), and can be done in about $n^2/2$ flops. After that this Barycentric form only requires about $4n$ flops per evaluation point. There is more to be said if you want to evaluate P for many points all at once... but that is another course (look up polynomial evaluation and the FFT and/or Cauchy matrices).