Computation of NURBS Basis Functions:

As with B-splines, the local support of NURBS basis functions suggest there is an efficient means of computing them. Indeed, supposing that

$$\xi_j \in [\xi^{(j)}_{l_j}, \xi^{(j)}_{l_j+1})$$

only basis functions $\{R_{i,p}\}_{i \in \mathcal{L}}$ are nonzero where $\mathcal{L}$ is the set:

$$\mathcal{L} = \{ i \in I : l_j - p_j \leq i_j \leq l_j \}$$

The pseudocode below finds this set $\mathcal{L}$ and computes the corresponding NURBS basis functions by calling Compute Spline Basis in each parametric direction.

function Compute NURBS Basis ($\vec{\xi}$)
begin
  for $j = 1, \ldots, d_s$
    call Compute Spline Basis (ξ_j) and set $l_j = l$ and
    $N^{(j)}_{l_j - k, p_j} = N_{l-k,p}$ for $k = 0, \ldots, p_j$
  endloop

  Set $\mathcal{L} = \{ i \in I : l_j - p_j \leq i_j \leq l_j \}$

  For $i \in \mathcal{L}$
    Set $N_{i,p} = \prod_{j=1}^{d_s} N^{(j)}_{i_j, p_j}$
  endloop

  Set $w = \sum_{i \in \mathcal{L}} w_i N_{i,p}$

  for $i \in \mathcal{L}$
    Set $R_{i,p} = (w_i N_{i,p}) / w$
  endloop

  return $\mathcal{L}$ and $\{R_{i,p}\}_{i \in \mathcal{L}}$
end

Using Compute NURBS Basis, we can then easily evaluate a NURBS geometry.

function Compute NURBS Geometry ($\vec{\xi}$)
begin
  call Compute NURBS Basis ($\vec{\xi}$)
  return $\sum_{i \in \mathcal{L}} \vec{P}_i R_{i,p}$
end