A Parallel Algorithm for Even N

As mentioned in Section 4, when the problem size N is even, a small modification to Algorithm 4 is required. This modification is shown in Algorithm 5, whose first loop is similar to the one in the algorithm initially shown, and can also be seen as a loop where each bead compares itself with subsequent beads in a circular fashion. Here we also model this circularity using modular arithmetic, so bead i in iteration s evaluates bead $reach(s)$, and is evaluated by bead $reached(s)$.

Algorithm 5: Proposed algorithm for calculating Symmetric Pairwise Interactions (SPI) for even N.

```
for (i = 0 to N-1){
    for (j = 1 to (N-2)/2)
        interactions += interact(obj[i], obj[(i+j)%N]);
    if (i < N/2)        /* only half beads perform one more iteration */
        interactions += interact(obj[i], obj[(i+N/2)%N]);
}
```

For odd number of beads, the problem was that $reach(s)$ eventually crossed with $reached(s)$, which was when interactions among beads began to be calculated twice (illustrated in Figure 1). For even N, what happens is that $reach(s)$ eventually becomes equal to $reached(s)$, which means that the same interaction is calculated twice in the same iteration, numbered $N/2$. This situation can be avoided by allowing only the first half of the bead vector to perform such iteration, which is the purpose of the **if** condition in Algorithm 5. We now again prove that this algorithm perform the same number of comparisons as the straightforward algorithm (that is, $N.(N-1)/2$) and that all of them are different.

Proposition 3. *The proposed algorithm evaluates $N.(N-1)/2$ interactions among beads, for even N.*

Proof. As seen in Algorithm 5, all N beads initially perform $(N-2)/2$ comparisons, which gives $N.(N-2)/2$ comparisons. Note that $(N-2)/2$ is an integer number because N is even. After that, $N/2$ beads perform one more iteration, which gives $N/2$ additional comparisons. Summing everything yields

$$\frac{N.(N-2)}{2} + \frac{N.(1)}{2} = \frac{N.(N-2+1)}{2} = \frac{N.(N-1)}{2}$$

which is the number of existent pairwise interactions. □

Proposition 4. *All $N.(N-1)/2$ interactions evaluated by the proposed algorithm are different from each other, for even N.*

Proof. Take two arbitrarily different beads i and j, with $i < j$. Each bead evaluates the interaction between itself and subsequent beads, so for beads i and j one side of the interactions they evaluate is inherently different, since $i \neq j$. Consequently, both beads would only evaluate the same interaction if it was the interaction among beads i and j themselves.

As seen in Proposition 2, bead i evaluates its interaction with bead j when $s = j - i$, and bead j evaluates bead i when $s = N - (j - i)$. Letting the distance between i and j be called $d = j - i$, then we have

$$\begin{aligned} s &= d && \text{bead i evaluates j} \\ s &= N - d && \text{bead j evaluates i} \end{aligned}$$

Looking back at Algorithm 5, the beads can perform up to $N/2$ iterations, so a hazard would occur if there existed a $d \in \{1, ..., N-1\}$ (valid distance between beads i and j) such that s was a valid iteration, that is,

$$s = d \in \{1, \ldots, N/2\} \tag{5}$$

$$s = N - d \in \{1, \ldots, N/2\} \tag{6}$$

Note that as d increases, $N - d$ decreases; taking $d = (N-2)/2$ satisfies (5), and (6) becomes

$$s = N - d = N - \frac{N-2}{2} = \frac{2.N - (N-2)}{2} = \frac{N+2}{2}$$

So (6) is still not satisfied. At this point, decreasing d would increase the value of (6) and it would remain not satisfied, so $d \leq (N-2)/2$ is proven to not satisfy both equations at the same time. What is left is to increase d, and the only possibility is to take $d = N/2$, which is the highest value of d that still satisfies (5). Equation (6) would then be satisfied because

$$s = N - d = N - \frac{N}{2} = \frac{N}{2} \in \{1, \ldots, N/2\}$$

This means that in iteration $s = N/2$, all pairs of beads i and j whose distance from each other is $N/2$ evaluate themselves at the same time, giving a duplicate interaction evaluation. However, Algorithm 5 doesn't have this problem precisely because in iteration $s = N/2$ it allows only a set of beads to execute, a set in which all beads are within a distance $d < N/2$ from each other; this can be done by taking the set of beads $\{0, 1, ..., (N-2)/2\}$ (first half of the bead vector), in which the largest distance is between beads 0 and $(N-2)/2$, giving a distance of $(N-2)/2 < N/2$.

Therefore, in summary, a hazard occurs if bead i evaluates bead j in iteration s_1 of the algorithm, and bead j evaluates bead i in iteration s_2. Algorithm 5 performs iterations $s \in \{1, ..., N/2\}$, and we have proved that the only hazard that can occur is that beads separated by a distance of $N/2$ evaluate each other at the same time in iteration $s = N/2$. Since Algorithm 5 only allows the first half of the bead vector to perform iteration $N/2$, no hazard occurs during the algorithm. Hence, all the evaluated interactions are different from each other, concluding the proof. □

Because the algorithm evaluates $N.(N-1)/2$ interactions, which is precisely the number of pairwise interactions that exist among beads, and all interactions are different from each other, then Algorithm 5 is correct.