Parallel Programming 2nd edition
ERRATA
As of June 3, 2011

The authors thank all those who reported mistakes.
Please report mistakes to abw@uncc.edu.

Page 64 line 19: “However in many problems, the data type of the data been sent is often the same throughout.” should read “However in many problems, the data type of the data being sent is often the same throughout.”

(We are very grateful to Michael Greifenkamp, University of Illinois, Urbana-Champaign for pointing out this mistake.)

Page 77 Problem 2.4: “... as given in Figures 2.14 and 2.16” should read “... as given in Figure 2.12”

Page 114, line 8: 
send(s2, P1};
should read
send(s2, P1);
i.e. the curly bracket should be a round bracket.

Page 118: Sequential Algorithm: “A more effective way is to divide the number by m/a ... ” should be “A more effective way is to divide the number by a/m .. ” i.e. m/a should be a/m (several places).

Page 147 - 148: Single Instance of Problem
The statement “... and two communications, one communication from the left and one communication from the right.” does not take into account that both communications will overlap and take place at the same time (although for different instances of the problem. Hence the equation:

\[ t_{\text{comm}} = 2(t_{\text{startup}} + t_{\text{data}}) \]

should read

\[ t_{\text{comm}} = t_{\text{startup}} + t_{\text{data}} \]
This correction then affects the subsequent equations on page 147 and page 148.

(We are very grateful to Professor Fernando G. Tinetti, Universidad Nacional de La Plata and Instituto Tecnológico de Buenos Aires for pointing out this mistake.)

Page 222: The line:

```c
} else vertex_queue == FALSE;
```

should read:

```c
} else vertex_queue = FALSE;
```

Page 256: The beginning brace of a structured block that follow a OpenMP pragma must start on a new line. Hence the correction below:

```c
#pragma omp parallel sections
{
  #pragma omp section
  structured_block
  #pragma omp section
  structured_block
  ...
}
```

Page 330: Corrections are shown in red:

First, $c[]$ will be used to hold the histogram of the sequence, that is, the number of each number. This can be computed in $\mathcal{O}(m + n)$ time with code such as:

```c
for (i = 1; i <= m; i++)
  c[i] = 0;
for (i = 1; i <= n; i++)
  c[a[i]]++;
```

... In the final stage of the algorithm, the numbers are placed in the sorted order in $\mathcal{O}(n)$ time, as described below:

```c
for (i = n; i >= 1; i--) {
  b[c[a[i]]] = a[i];
  c[a[i]]--; // done to ensure stable sorting
}
```

The complete code has $\mathcal{O}(n + m)$ sequential time complexity.
Page 350: The line:

\[ \text{recv(&b,P}_{i-1,j}); /* receive from right */ } \]

should read:

\[ \text{recv(&b,P}_{i-1,j}); /* receive from above */ } \]

Page 355:

\[ t_{\text{comp}} = 2 \sum_{j=1}^{n-1} (n-j+2) = \frac{(n+3)(n+2)}{2} - 3 = O(n^2) \]

should read:

\[ t_{\text{comp}} = 2 \sum_{j=1}^{n-1} (n-j+2) = (n-1)(n+4) = O(n^2) \]

(We are very grateful to Gaoyuan Huang for pointing out this mistake.)

Page 403: Figure 12.36 should look like:

![Figure 12.36](image.png)