

Lin/Snyder, *Principles of Parallel Programming*, Figure 7.8, Fixes

```
101 do
102 { /*
103  * Send data to four neighbors */
104  int num_requests=0;
105  if(row!=Top)          /* Send North */
106  {
107      MPI_Isend(&val[1][1], Width-2, MPI_FLOAT,
108              NorthPE(myID), tag, MPI_COMM_WORLD, &requests[0]);
108.5  num_requests++;
109  }
110
111  if(col!=Right)       /* Send East */
112  {
113      for(i=1; i<Height-1; i++)
114      {
115          buffer1[i-1]=val[i][Width-2];
116      }
117      MPI_Isend(buffer1, Height-2, MPI_FLOAT,
118              EastPE(myID), tag, MPI_COMM_WORLD, &requests[1]);
118.5  num_requests++;
119  }
120
121  if(row!=Bottom)     /* Send South */
122  {
123      MPI_Isend(&val[Height-2][1], Width-2, MPI_FLOAT,
124              SouthPE(myID), tag, MPI_COMM_WORLD, &requests[2]);
124.5  num_requests++;
125  }
126
127  if(col!=Left)       /* Send West */
128  {
129      for(i=1; i<Height-1; i++)
130      {
131          buffer2[i-1]=val[i][1];
132      }
133      MPI_Isend(buffer2, Height-2, MPI_FLOAT,
134              WestPE(myID), tag, MPI_COMM_WORLD, &requests[3]);
134.5  num_requests++;
135  }
136
137  /*
138  * Receive messages
139  */
140  if(row!=Top)        /* Receive from North */
141  {
142      MPI_Irecv(&val[0][1], Width-2, MPI_FLOAT,
143              NorthPE(myID), tag, MPI_COMM_WORLD, &requests[4]);
143.5  num_requests++;
144  }
145
146  if(col!=Right)     /* Receive from East */
147  {
```

```

148     MPI_Irecv(&buffer3, Height-2, MPI_FLOAT,
149             EastPE(myID), tag, MPI_COMM_WORLD, &requests[5]);
149.5    num_requests++;
150    for(i=1; i<Height-1; i++)
151    {
152        val[i][Width-1]=buffer3[i-1];
153    }
154    }
155
156    if(row!=Bottom)        /* Receive from South */
157    {
158        MPI_Irecv(&val[Height-1][1], Width-2, MPI_FLOAT,
159                SouthPE(myID), tag, MPI_COMM_WORLD, &requests[6]);
159.5    num_requests++;
160    }
161
162    if(col!=Left)          /* Receive from West */
163    {
164        MPI_Irecv(&buffer4, Height-2, MPI_FLOAT,
165                WestPE(myID), tag, MPI_COMM_WORLD, &requests[7]);
165.5    num_requests++;
166    for(i=1; i<Height-1; i++)
167    {
168        val[i][0]=buffer4[i-1];
169    }
170    }
171
172    delta=0.0; /* Calculate average, delta for all points */
173    for(i=2; i<Height-2; i++)
174    {
175        for(j=2; j<Width-2; j++)
176        {
177            average=(val[i-1][j]+val[i][j+1]+
178                    val[i+1][j]+val[i][j-1])/4;
179            delta=Max(delta, Abs(average - val[i][j]));
180            new[i][j]=average;
181        }
182    }
183    MPI_Waitall(num_requests, requests, status);
184
185    if(col!=Right)        /* Receive from East */
186    for(i=1; i<Height-1; i++)
187    {
188        val[i][Width-1]=buffer3[i-1];
189    }
190
191    if(col!=Left)        /* Receive from West */
192    for(i=1; i<Height-1; i++)
193    {
194        val[i][0]=buffer4[i-1];
195    }
196
197    /* update top and bottom edges, including corners */
198    for(j=1; j<Width-1; j++)
199    {
200        i=1;
201        average=(val[i-1][j]+val[i][j+1]+
202                val[i+1][j]+val[i][j-1])/4;
203        delta=Max(delta, Abs(average-val[i][j]));
204        new[i][j]=average;
205
206        i=Height-2;
207        average=(val[i-1][j]+val[i][j+1]+
208                val[i+1][j]+val[i][j-1])/4;

```

```
197     delta=Max(delta, Abs(average-val[i][j]));
198     new[i][j]=average;
199 }
200
201 /* update left and right edges, excluding corners */
202 for(i=2; i<Height-2; i++)
203 {
204     j=1;
205     average=(val[i-1][j]+val[i][j+1]+
206             val[i+1][j]+val[i][j-1])/4;
207     delta=Max(delta, Abs(average - val[i][j]));
208     new[i][j]=average;
209
210     j=Width-2;
211     average=(val[i-1][j]+val[i][j+1]+
212             val[i+1][j]+val[i][j-1])/4;
213     delta=Max(delta, Abs(average-val[i][j]));
214     new[i][j]=average;
215 }
216 /* Find maximum diff */
217 MPI_Reduce(&delta, &globalDelta, 1, MPI_FLOAT, MPI_MAX,
218           RootProcess, MPI_COMM_WORLD);
219 Swap(val, new);
220 } while(globalDelta >= THRESHOLD);
```