pardil (1 call, 1518.930 sec)
*Generated 08-May-2010 07:53:28 using cpu time.*
M-function in file /Users/jhd/Dropbox/uni/15. semester/msc-vc/svn/msc-vc/code/pardil.m
Copy to new window for comparing multiple runs

Refresh

Show parent functions  Show busy lines  Show child functions
Show M-Lint results  Show file coverage  Show function listing

**Parents** (calling functions)
No parent

**Lines where the most time was spent**

| Line Number | Code | Calls | Total Time | % Time | Time Plot |
|---|---|---|---|---|---|
| 34 | `temp = X - (sx * (i-is).*(i-is...` | 262144 | 1421.136 s | 93.6% | |
| 35 | `output(i,j) = max(temp(:));` | 262144 | 96.248 s | 6.3% | |
| 36 | `end` | 262144 | 1.516 s | 0.1% | |
| 14 | `js = zeros(m,n);` | 1 | 0.020 s | 0.0% | |
| 17 | `is(i,:) = i;` | 512 | 0.010 s | 0.0% | |
| All other lines | | | 0 s | 0% | |
| Totals | | | 1518.930 s | 100% | |

**Children** (called functions)
No children

**M-Lint results**
No M-Lint messages.

**Coverage results**
[ Show coverage for parent directory ]

| | |
|---|---|
| Total lines in function | 39 |
| Non-code lines (comments, blank lines) | 16 |
| Code lines (lines that can run) | 23 |
| Code lines that did run | 20 |
| Code lines that did not run | 3 |
| Coverage (did run/can run) | 86.96 % |

**Function listing**
Color highlight code according to time

```
time    calls  line
                1 function [output] = pardil(X,scaling)
                2
```

```
            1     3 sx = scaling(1);
            1     4 sy = scaling(2);
              5
            1     6 if (sx < 0) || (sy < 0)
              7     disp('The scaling factor must be >= 0.');
              8     return;
              9 end
             10
            1    11 [m,n] = size(X);
             12
            1    13 is = zeros(m,n);
 0.02       1    14 js = zeros(m,n);
             15
            1    16 for i=1:m
 0.01     512    17     is(i,:) = i;
          512    18 end
             19
            1    20 for j=1:n
          512    21     js(:,j) = j;
          512    22 end
             23
             24 % temp = zeros(m,n);
            1    25 output = zeros(m,n);
             26
            1    27 for i=1:m
          512    28     for j=1:n
             29 %         for i1=1:m
             30 %             for j1=1:n
             31 %                 temp(i1,j1) = X(i1,j1) - (sx * (i-i1)*(i-i1)
             32 %             end
             33 %         end
1421.14  262144    34         temp = X - (sx * (i-is).*(i-is) + sy * (j-js).*(j-js)
96.25    262144    35         output(i,j) = max(temp(:));
 1.52    262144    36     end
          512    37 end
             38
            1    39 end
```