

pardil (1 call, 36318.617 sec)

Generated 08-May-2010 19:08:19 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
<a href="#">31</a>	temp(i1,j1) = X(i1,j1) - (sx *...	4.294967e+09	18091.435 s	49.8%	
<a href="#">32</a>	end	4.294967e+09	18073.423 s	49.8%	
<a href="#">30</a>	for j1=1:n	16777216	74.826 s	0.2%	
<a href="#">33</a>	end	16777216	71.159 s	0.2%	
<a href="#">35</a>	output(i,j) = max(temp(:));	65536	6.892 s	0.0%	
All other lines			0.882 s	0.0%	
Totals			36318.617 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)	19
Code lines (lines that can run)	20
Code lines that did run	17
Code lines that did not run	3
Coverage (did run/can run)	85.00 %

**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
13 % is = zeros(m,n);
14 % js = zeros(m,n);
15 %
16 % for i=1:m
17 %     is(i,:) = i;
18 % end
19 %
20 % for j=1:n
21 %     js(:,j) = j;
22 % end
23
1 24 temp = zeros(m,n);
1 25 output = zeros(m,n);
26
1 27 for i=1:m
256 28     for j=1:n
0.45 65536 29         for il=1:m
74.83 16777216 30             for j1=1:n
18091.44 4.294967e+09 31                 temp(il,j1) = X(il,j1) - (sx * (i-il))*
18073.42 4.294967e+09 32                 end
71.16 16777216 33             end
34 %                 temp = X - (sx * (i-is).*(i-is) + sy * (j-js).*(j-js)
6.89 65536 35                 output(i,j) = max(temp(:));
0.42 65536 36             end
256 37         end
38
1 39     end
```