SFDTD3DfunC (Calls: 6000, Time: 2354.153 s)

Generated 10-Oct-2017 21:25:12 using performance time.
function in file E:\GIT\IndiEngiSchola\Matlab\SFDTD\SFDTD3DfunC.m
Copy to new window for comparing multiple runs

Refresh

☑ Show parent functions ☑ Show busy lines ☑ Show child functions
☑ Show Code Analyzer results ☑ Show file coverage ☑ Show function listing

Parents (calling functions)

Function Name	Function Type	Calls
SFDTD3Dtesting	script	6000

Lines where the most time was spent

Line Number	Code	Calls	Total Time
79	`p(i,i1,i2) = p(i,i1,i2) - pCx*...`	207507525	135.894 s
81	`- pCz*(uz(i,i1, i2 + 1) - uz(i...`	207507525	134.259 s
80	`- pCy*(uy(i+1, i1, i2) - uy(i,...`	207507525	134.054 s
78	`if(idx(i, i1, i2) > 0)`	207518850	132.831 s
10	`ux(i,i1,i2) = ux(i,i1,i2) - uC...`	202567675	131.642 s
All other lines			1685.473 s
Totals			2354.153 s

Children (called functions)
No children

Code Analyzer results

Line number	Message
2	`The value assigned to variable 'sizemat' might be unused.`
74	`IF might not be aligned with its matching END (li 103).`

Coverage results
Show coverage for parent directory

	% Time	Time Plot
	5.8%	
	5.7%	
	5.7%	
	5.6%	
	5.6%	
	71.6%	
	100%	

ne

Total lines in function	104
Non-code lines (comments, blank lines)	42
Code lines (lines that can run)	62
Code lines that did run	62
Code lines that did not run	0
Coverage (did run/can run)	100.00 %

Function listing

Color highlight code according to time

time	Calls	line	
		1	function [p, ux, uy, uz] = SFDTD3DfunC(p,
0.02	6000	2	sizemat = size(p);
1.13	6000	3	if(size(idx(idx > 0),1) > 100)
		4	
		5	% mat(col, row)
0.03	5989	6	for i = 1 : size(ux,1)-1
0.31	197637	7	for i1 = 2 : size(ux,2)-2
11.40	8103117	8	for i2 = 1 : size(ux,3)-1
127.04	202577925	9	if(idx(i, i1,i2) > 0)
131.64	202567675	10	ux(i,i1,i2) = ux(i,i1,
120.91	202567675	11	end
121.73	202577925	12	end
4.94	8103117	13	end
0.14	197637	14	end
		15	
0.01	5989	16	for i = 2 : size(uy,1)-2
0.29	191648	17	for i1 = 1 : size(uy,2)-1
11.24	8049216	18	for i2 = 1 : size(uy,3)-1
127.11	201230400	19	if(idx(i, i1,i2) > 0)
131.24	201219100	20	uy(i,i1,i2) = uy(i,i1,
120.22	201219100	21	end
121.11	201230400	22	end
4.91	8049216	23	end
0.13	191648	24	end
0.02	5989	25	for i = 1 : size(uz,1)-1
0.30	197637	26	for i1 = 1 : size(uz,2)-1
11.54	8300754	27	for i2 = 2 : size(uz,3)-2
125.90	199218096	28	if(idx(i, i1,i2) > 0)
130.07	199207224	29	uz(i,i1,i2) = uz(i,i1,
119.04	199207224	30	end
120.13	199218096	31	end
5.10	8300754	32	end

```
pCx, pCy, pCz, ux, uy, uz, uCx, uCy, uCz, Rx, Ry, Rz, ZL, ZR, ZT, Z
```

```
,i2) - uCx*(p(i,i1,i2)-p(i,i1-1,i2));
```

```
,i2) - uCx*(p(i,i1,i2)-p(i-1,i1,i2));
```

```
,i2) - uCx*(p(i,i1,i2)-p(i,i1,i2-1));
```

```
B, ZF, ZG, idx)
```

```
  0.13    197637    33      end
                    34      % if length(idx) > 10
                    35      % %     %  mat(col, row)
                    36      % [suby, subx, subz] = ind2sub(sizemat
                    37      % suby = suby + 1;
                    38      % subx = subx + 1;
                    39      % subz = subz + 1;
                    40      % for i = 1 : length(suby)
                    41      %     ux(suby,subx,subz) = ux(suby,sub
                    42      %     uy(suby,subx,subz) = uy(suby,sub
                    43      %     uz(suby,subx,subz) = uz(suby,sub
                    44      % end
< 0.01         11   45  else
                    46      % update the non-boundary condition no
< 0.01         11   47      ux(:, 2:end-1, :) = ux(:, 2:end-1,:) -
< 0.01         11   48      uy(2:end-1, :, :) = uy(2:end-1, :, :)
< 0.01         11   49      uz(:, :, 2:end-1) = uz(:, :, 2:end-1)
< 0.01       6000   50  end
                    51
                    52  % update the velocity at the right wall
  0.18       6000   53  ux(:, end, :) = ((Rx - ZR)/(Rx + ZR))*ux(:
             6000   54      + (2/(Rx + ZR))*p(:, end, :);
                    55
                    56  %update the velocity at the left wall
  0.07       6000   57  ux(:, 1, :) = ((Rx - ZL)/(Rx + ZL))*ux(:,
                    58
                    59  %update the velocity at the top wall
  0.19       6000   60  uy(end, :, :) = ((Ry - ZF)/(Ry + ZF))*uy(:
             6000   61      + (2/(Ry + ZT))*p(end, :, :);
                    62
                    63  %update the velocity at the bottom wall
  0.12       6000   64  uy(1, :, :) = ((Ry - ZB)/(Ry + ZB))*uy(1,
                    65
                    66  %update the velocity at the ceiling
  0.09       6000   67  uz(:, :, end) = ((Rz - ZT)/(Rz + ZT))*uz(:
             6000   68      (2/(Rz + ZT))*p(:, :, end);
                    69
                    70  %update the velocity at the floor
  0.07       6000   71  uz(:, :, 1) = ((Rz - ZG)/(Rz + ZG))*uz(:,
             6000   72      (2/(Rz + ZG))*p(:, :, 1);
                    73
  1.28       6000   74  if(size(idx(idx > 0),1) > 10)
  0.03       5989   75      for i = 1 : size(p,1)-1
  0.32     197637   76          for i1 = 1 : size(p,2)-1
 12.21    8300754   77              for i2 = 1 : size(p,3)-1
```

```
t,idx);
```

```
ox,subz) - uCx*(p(suby,subx,subz)-p(suby,subx-1,subz));
ox,subz) - uCx*(p(suby,subx,subz)-p(suby-1,subx,subz));
ox,subz) - uCx*(p(suby,subx,subz)-p(suby,subx,subz-1));
```

```
odes for velocity
- uCx*(p(:, 2:end,:) - p(:, 1:end-1, :));
 - uCy*(p(2:end, :, :) - p(1:end-1, :, :));
 - uCz*(p(:, :, 2:end) - p(:, :, 1:end-1));
```

```
:, end, :) ...
```

```
 1, :) - (2/(Rx + ZL))*p(:, 1, :);
```

```
end, :, :) ...
```

```
 :, :) - (2/(Ry + ZB))*p(1, :, :);
```

```
:, :, end) + ...
```

```
 :, 1) - ...
```

132.83	207518850	78	if(idx(i, i1, i2) > 0)
404.21	207507525	79	p(i,i1,i2) = p(i,i1,i2
	207507525	80	- pCy*(uy(i+1, i1,
	207507525	81	- pCz*(uz(i,i1, i2
124.41	207507525	82	end
125.03	207518850	83	end
5.08	8300754	84	end
0.15	197637	85	end
		86	% if length(idx) > 100
		87	% p(idx) = p(idx) - pCx*(ux(:, idx
		88	% - pCy*(uy(idx+1, :, :) - uy
		89	% - pCz*(uz(:, :, idx+1) - uz
		90	% if length(idx) > 100
		91	% % mat(col, row)
		92	% % for i = 1 : length(suby)
		93	%
		94	% p(suby,subx,subz) = p(su
		95	% - pCx*(ux(suby,subx-
		96	% - pCy*(uy(suby+1,sub
		97	% - pCz*(uz(suby,subx,
		98	% end
< 0.01	11	99	else
< 0.01	11	100	p = p - pCx*(ux(:, 2:end, :) - ux
	11	101	- pCy*(uy(2:end, :, :) - uy(1
	11	102	- pCz*(uz(:, :, 2:end) - uz(:,
< 0.01	6000	103	end
0.04	6000	104	end

```
2) - pCx*(ux(i, i1 + 1, i2) - ux(i, i1, i2))...
, i2) - uy(i, i1, i2))...
2 + 1) - uz(i, i1, i2));
```

```
x+1, :) - ux(:, idx, :))...
(idx, :, :))...
(:, :, idx));
```

```
uby,subx,subz)...
+1,subz) - ux(suby,subx,subz))...
ox,subz) - uy(suby,subx,subz))...
,subz+1) - uz(suby,subx,subz));
```

```
(:, 1:end-1, :))...
:end-1, :, :))...
, :, 1:end-1));
```