

```

function SimulateMLApair
% authors:      Edwin van der Pol, Frank Coumans
% last edited : January 6th 2012
% contact:     f.a.w.coumans.at.gmail.com

%define global variables
global fF ROC pL N0 Nya Ny Nalpha alphaMaxRad y dMLA nFusedSilica spectrum c

% Original values for reference; fF = 0.020; ROC = 0.0022; dMLA = 0.005;
% pL = 0.0003; N0 = 16; Nya = 256; Ny = 1024; Nalpha = 250; alphaMax = 0.7;
% dye = 'HG'; TargetSize = 0.8;

%define variables
fF = 0.020; %focus distance objective [m]
ROC = 0.0022; %radius of curvature ML's [m]
dMLA = 0.005; %distance between MLA's (fL<dMLA<2fL) [m], with the original
values fL = 0.00475
pL = 0.0003; %MLA pitch [m]
N0 = 16; %number of ML's (typical 16)
Nya = 256; %number of datapoints on ya axis (needs to be 256 or higher power
of 2 for original values)
Ny = 1024; %number of datapoints on y axis (typical 1024)
Nalpha = 250; %alpha integration number (50 for fast simulation, 250 for good
simulation)
alphaMax = 0.7; %divergence [degrees at half width]
dye = 'HG'; %choose dye, use: APC, FITC, DAPI, PE, or HG
TargetSize = 0.8; %Size of pickup area [mm]

%define filter spectra
APC = [0.0008 0.0016 0.0080 0.0152 0.0715 0.1384 0.2690 0.3920 0.5079 0.7401
0.8389 0.8237 1.0000 0.9942 0.9662 0.9384 0.9121 0.8860 0.8826 0.8824 0.8822
0.8820 0.8690 0.8418 0.8287 0.8285 0.8283 0.8283 0.8281 0.8279 0.8279 0.8277
0.8275 0.8273 0.8273 0.6811 0.6810 0.5837 0.2918 0.0972 0.0486 0.0087 0.0039
0.0007 0.0003; 598 599 600 601 602 603 604 605 606 607 608 609 610 611 612
613 614 615 616 617 618 619 620 621 622 623 624 625 626 627 628 629 630 631
632 633 634 635 636 637 638 639 640 641 642];
FITC = [0.0006 0.0012 0.0060 0.0122 0.0627 0.1267 0.5066 0.8985 0.9243 0.9426
0.9425 0.9453 0.9726 1.0000 0.9763 0.9483 0.8674 0.7699 0.7485 0.7697 0.7917
0.8144 0.8310 0.4747 0.1190 0.0612 0.0126 0.0067 0.0013 0.0006; 463 464 465
466 467 468 469 470 471 472 473 474 475 476 477 478 479 480 481 482 483 484
485 486 487 488 489 490 491 492];
DAPI = [0.0002 0.0002 0.0003 0.0002 0.0011 0.0010 0.0019 0.0019 0.0102 0.0114
0.0118 0.0121 0.0263 0.0294 0.0633 0.0651 0.1120 0.1763 0.3207 0.5026 0.6030
0.7332 0.8882 1.0000 0.9998 0.9821 0.8129 0.6452 0.5356 0.4259 0.0739 0.0317
0.0056 0.0025 0.0005 0.0002 0.0002 0.0002 0.0002 0.0001; 341 342 343 344 345
346 347 348 349 350 351 352 353 354 355 356 357 358 359 360 361 362 363 364
365 366 367 368 369 370 371 372 373 374 375 376 377 378 379 380];
PE = [0.0001 0.0004 0.0010 0.0061 0.0151 0.0540 0.0922 0.1753 0.3232 0.5751
0.8269 0.9218 1.0000 0.9350 0.8055 0.6603 0.5039 0.3673 0.2482 0.0250 0.0102
0.0016 0.0006 0.0001; 534 535 536 537 538 539 540 541 542 543 544 545 546 547
548 549 550 551 552 553 554 555 556 557];
HG = [0.459146406 0.434634869 0.398633356 0.371746935 0.361249078 0.354843710
0.365074936 0.375430177 0.397821059 0.420218142 0.456070837 0.495098313
0.560032492 0.642595383 0.689069951 0.669618220 0.616955311 0.425935227
0.271685548 0.256741758 0.241897179 0.228695798 0.215494416 0.203700603

```

0.192037006 0.185271996 0.179982762 0.187863906 0.204221466 0.227778088  
0.262384434 0.298906809 0.346044856 0.393182903 0.430647792 0.467176368  
0.421743525 0.340817630 0.273682187 0.215047963 0.188812620 0.188781616  
0.178141141 0.150138587 0.126290530 0.112803914 0.101568168 0.101549566  
0.101530963 0.107551885 0.113610010 0.117510278 0.120840077 0.131461949  
0.146839791 0.158125143 0.162664087 0.186716769 0.293784995 0.400853222  
0.502557806 0.603040844 0.733244043 0.888157201 1 0.999826379 0.982067452  
0.812892585 0.645199695 0.535551959 0.425904223 0.369706891 0.316578926  
0.281259495 0.249344271 0.226624750 0.207495458 0.190263594 0.174377290  
0.159718734 0.146529754 0.137557279 0.137532476 0.135907882 0.124963571  
0.114081267 0.107868123 0.101661179 0.0960805100 0.0905928530 0.0872630540  
0.0847765560 0.0823706680 0.0800329880 0.0805042440 0.0851982070 0.0924778790  
0.110199602 0.127921325 0.250782844 0.379869908 0.488655741 0.577177546  
0.653390876 0.691823080 0.726745664 0.687116717 0.647487769 0.597447774  
0.546235839 0.408765370 0.244811528 0.176138301 0.132193637 0.109505119  
0.0942202880 0.0855826530 0.0807584750 0.0772240510 0.0750351890 0.0742724980  
0.0764055530 0.0785572110 0.0808266830 0.0830961550 0.0931909640 0.103670220  
0.116276330 0.129682336 0.147404059 0.169447700 0.195348203 0.232360437  
0.269428478 0.323071104 0.376707530 0.498508721 0.643227859 0.677102517  
0.569637443 0.466674108 0.372199590 0.289804119 0.238313150 0.193091132  
0.172659685 0.152302646 0.132832313 0.113361981 0.0989576550 0.0846773440  
0.0788982520 0.0743903120 0.0689336580 0.0631111610 0.0578777340 0.0530411540  
0.0485828200 0.0445709400 0.0414767690 0.0403048290 0.0391514900 0.0380477580  
0.0369564270 0.0369502270 0.0369440260 0.0378307320 0.0389034610 0.0392817060  
0.0392693040 0.0398025680 0.0409435050 0.0417558020 0.0417496020 0.0418736160  
0.0430827610 0.0442981070 0.0432501810 0.0420100330 0.0384260030 0.0341040860  
0.0331553720 0.0340978850 0.0350714020 0.0360759220 0.0368138100 0.0368014090  
0.0368944200 0.0379671480 0.0390336760 0.0412473410 0.0435850210 0.0451786120  
0.0464621660 0.0464001590 0.0450856010 0.0433803970 0.0409745090 0.0388414530  
0.0377501220 0.0366153870 0.0346187470 0.0326159070 0.0308300930 0.0290690820  
0.0280335580 0.0272336630 0.0264585700 0.0257082800 0.0252866290 0.0252804290  
0.0252742280 0.0252680270 0.0252680270 0.0252618260 0.0252556260 0.0252494250  
0.0252494250 0.0252432240 0.0252370230 0.0256524730 0.0263841610 0.0268368150  
0.0268306140 0.0269856330 0.0277607260 0.0285420190 0.0293667180 0.0301976180  
0.0310657220 0.0319462270 0.0348605760 0.0389220630 0.0428719360 0.0466667910  
0.0520490350 0.0639172570 0.0757916800 0.0936188160 0.111923408 0.143410781  
0.181749974 0.287218409 0.510978415 0.734732221 0.819043721 0.888554049  
0.830831334 0.715726944 0.586720489 0.447737039 0.326338895 0.220572825  
0.155589040 0.126668775 0.102337060 0.0835984150 0.0703598290 0.0664533610  
0.0634211980 0.0634087960 0.0633963950 0.0633839930 0.0633777930 0.0666827890  
0.0704652420 0.0794997240 0.0910827120 0.101543365 0.110515840 0.122625891  
0.146033695 0.169447700 0.196805377 0.224330475 0.339496872 0.488816960  
0.586931314 0.586825902 0.561006009 0.408114292 0.270836046 0.218061524  
0.167761098 0.142542677 0.117324255 0.107936331 0.0986104140 0.0886334180  
0.0785510100 0.0714697620 0.0654240380 0.0609719040 0.0575428940 0.0542999050  
0.0512677420 0.0483533930 0.0456746720 0.0432439810 0.0420286350 0.0408132890  
0.0385934230 0.0363735580 0.0351892160 0.0341846950 0.0332173790 0.0322748670  
0.0313571570 0.0304642500 0.0299123840 0.0299061830 0.0297325620 0.0288954620  
0.0280645620 0.0272770680 0.0264957740 0.0263965620 0.0263903620 0.0263841610  
0.0263779600 0.0259873130 0.0251750160 0.0247843690 0.0247781680 0.0247719680  
0.0247719680 0.0247657670 0.0247595660 0.0247595660 0.0247533650 0.0247471650  
0.0247409640 0.0247409640 0.0247347630 0.0247285620 0.0247285620 0.0247223620  
0.0247161610 0.0247161610 0.0247099600 0.0247037600 0.0246975590 0.0246975590  
0.0246913580 0.0246851570 0.0246851570 0.0246789570 0.0246727560 0.0246665550  
0.0246665550 0.0246603540 0.0246541540 0.0246541540 0.0246479530 0.0246417520  
0.0246417520 0.0246355510 0.0246293510 0.0246231500 0.0246231500 0.0246169490  
0.0246107480 0.0246107480 0.0246045480 0.0245983470 0.0245983470 0.0252990310

```

0.0260121160 0.0261547330 0.0261485330 0.0261423320 0.0261361310 0.0261361310
0.0261299300 0.0261237300 0.0261175290 0.0261919380 0.0269484280 0.0277049190
0.0277669260 0.0277607260 0.0277607260 0.0277545250 0.0277483240 0.0277421240
0.0280521610 0.0288582570 0.0294969340 0.0294969340 0.0294907330 0.0294845320
0.0294783320 0.0294721310 0.0294659300 0.0300425990 0.0308983020 0.0313323540
0.0313261530 0.0313199520 0.0313137510 0.0311897370000000; 300 301 302 303
304 305 306 307 308 309 310 311 312 313 314 315 316 317 318 319 320 321 322
323 324 325 326 327 328 329 330 331 332 333 334 335 336 337 338 339 340 341
342 343 344 345 346 347 348 349 350 351 352 353 354 355 356 357 358 359 360
361 362 363 364 365 366 367 368 369 370 371 372 373 374 375 376 377 378 379
380 381 382 383 384 385 386 387 388 389 390 391 392 393 394 395 396 397 398
399 400 401 402 403 404 405 406 407 408 409 410 411 412 413 414 415 416 417
418 419 420 421 422 423 424 425 426 427 428 429 430 431 432 433 434 435 436
437 438 439 440 441 442 443 444 445 446 447 448 449 450 451 452 453 454 455
456 457 458 459 460 461 462 463 464 465 466 467 468 469 470 471 472 473 474
475 476 477 478 479 480 481 482 483 484 485 486 487 488 489 490 491 492 493
494 495 496 497 498 499 500 501 502 503 504 505 506 507 508 509 510 511 512
513 514 515 516 517 518 519 520 521 522 523 524 525 526 527 528 529 530 531
532 533 534 535 536 537 538 539 540 541 542 543 544 545 546 547 548 549 550
551 552 553 554 555 556 557 558 559 560 561 562 563 564 565 566 567 568 569
570 571 572 573 574 575 576 577 578 579 580 581 582 583 584 585 586 587 588
589 590 591 592 593 594 595 596 597 598 599 600 601 602 603 604 605 606 607
608 609 610 611 612 613 614 615 616 617 618 619 620 621 622 623 624 625 626
627 628 629 630 631 632 633 634 635 636 637 638 639 640 641 642 643 644 645
646 647 648 649 650 651 652 653 654 655 656 657 658 659 660 661 662 663 664
665 666 667 668 669 670 671 672 673 674 675 676 677 678 679 680 681 682 683
684 685 686 687 688 689 690 691 692 693 694 695 696 697 698 699 700];
LED = [1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1; 490 491
492 493 494 495 496 497 498 499 500 501 502 503 504 505 506 507 508 509 510
511 512 513 514 515 516 517 518 519];

```

```

%refractive index fused silica

```

```

nFusedSilica = [300*10^-9 320*10^-9 360*10^-9 400*10^-9 450*10^-9 500*10^-9
550*10^-9 590*10^-9 600*10^-9 650*10^-9 700*10^-9; 1.48779 1.48274 1.47529
1.47012 1.46557 1.46233 1.46008 1.45846 1.45804 1.45653 1.45529];

```

```

%start calculations

```

```

alphaMaxRad = alphaMax*pi/180; %divergence [rad at half width]

```

```

%resize filter spectra to size APC

```

```

FITClambda = linspace(min(FITC(2,:)),max(FITC(2,:)),42);
FITC = interp1(FITC(2,:),FITC(1,:),FITClambda);
FITC(2,:) = FITClambda;
DAPIlambda = linspace(min(DAPI(2,:)),max(DAPI(2,:)),42);
DAPI = interp1(DAPI(2,:),DAPI(1,:),DAPIlambda);
DAPI(2,:) = DAPIlambda;
PElambda = linspace(min(PE(2,:)),max(PE(2,:)),42);
PE = interp1(PE(2,:),PE(1,:),PElambda);
PE(2,:) = PELambda;
HGLambda = linspace(min(HG(2,:)),max(HG(2,:)),42);
HG = interp1(HG(2,:),HG(1,:),HGLambda);
HG(2,:) = HGLambda;

```

```

switch dye %choose the corresponding spectrum
    case 'APC' ; spectrum = APC;
    case 'FITC'; spectrum = FITC;
    case 'DAPI'; spectrum = DAPI;
    case 'PE';   spectrum = PE;
    case 'HG';   spectrum = HG;
    case 'LED';  spectrum = LED;
    otherwise
        return
    end

%Calculate intensity distribution for wavelengths separately, then average.
Nlambda = length(spectrum); %number of lambda calculations
for c = 1:Nlambda
    lambda = spectrum(2,c)*10^-9; %define lambda
    disp([num2str(Nlambda-c+1) ', lambda: ' num2str(lambda*1e9,'%3.0f') '
nm']) %progress indicator

    lambdaResult = intalpha(lambda); %perform calculations for a specific
lambda

    if (c == 1) ; yFinal = y; end %create final y-scale. The scale of the
FFT output depends on lambda, therefore one final fixed scale is needed. This
will be the scale of the highest lambda.
    lambdaResultScaled (c,:) = interp1(y,lambdaResult,yFinal); %make an
interpolation for y,lambdaResult on the final fixed scale (yFinal)
end

lambdaResultScaled(isnan(lambdaResultScaled))=0; %replace all NAN with 0;
lambdaSum = sum(lambdaResultScaled,1)./Nlambda; %take the average of all the
lambda results
lambdaResultNorm = 1./max(lambdaSum)*lambdaSum; %normalize the graph

yFinal=yFinal*1000; %change units of y-scale from [m] to [mm]

%Calculate and display efficiency in 2D (assuming square pickup area) and
variation in 1D
IntTarget=0;
for i = find(yFinal>-TargetSize/2,1) : find(yFinal>TargetSize/2,1)
    IntTarget(i+1-find(yFinal>-TargetSize/2,1)) = lambdaResultNorm (i);
end
IntTotal = lambdaResultNorm; IntTotal(isnan(IntTotal))=[];
disp(['target_eff = ' num2str((sum(IntTarget)/sum(IntTotal))^2*100,'%3.1f')
'%'])
disp(['cv = ' num2str(std(IntTarget)/mean(IntTarget)*100,'%3.2f') '%'])

%plot the graph, second pair of variables indicate size pick up area
plot(yFinal,lambdaResultNorm,[-1 -1 1 1]*TargetSize/2,[0 1 1 0]);
YscaleMin = yFinal(find(lambdaResultNorm>0.05, 1, 'first'))*1.05; %restrict
the y-scale to useful range
axis([YscaleMin -YscaleMin 0 1])
xlabel('y[mm]')
ylabel('Relative Intensity')
end

```

```

function Isample = integrand(alpha,lambda)
% System diagram, '|' is a lens
%
%
%           ya|           yaa|           y
%           |           |
%U          |U*TL      UL1|UL1*TL  UL1bound|
Isample
%           |           |
%           |           |
%-----
-
% Distance between second MLA and field lens is assumed negligible

%define global variables
global fF fL ROC pL N0 Nya Ny y ya SN0 dMLA nFusedSilica spectrum c UL
UL1bound

%start calculations
fL = ROC/(interp1(nFusedSilica(1,:),nFusedSilica(2,:),lambda)-1); %calculate
the focuslength of the ML's for this wavelength
s = spectrum(1,c); %spectral distribution of the specified filter

dya = pL/Nya; %calculate the stepsize at zL1
ya = (-Nya/2:Nya/2-1) * dya; %create y' array
yaa = (-Nya:Nya-1) * lambda*fL/(2*Nya*dya); %determine the corresponding
Fourier scale at y"

U = s*exp(1i*2*pi*ya*sin(alpha)/lambda); %illumination impinging on the
array. The spatial distribution of the emitter is not taken into account.
d = exp((2*pi*1i/(lambda*fL))*(1-dMLA/fL)*yaa.^2); %quadratic phase factor
that compensates for the distance dMLA between L1 and L2 (fL<=dMLA<2fL)

UL_int = [zeros(1,Nya/2) U zeros(1,Nya/2)]; % pad with zeros to apply rect
function
UL1 = d.*fftshift(fft(UL_int)); %Fourier transformation performed on
transmitted field immediately behind the array

%Rect on yaa, is not nescessary for dMLA ~ fL
lbound = find(yaa>-pL/2,1,'first');
rbound = find(yaa>pL/2,1,'first');
for k = 1:rbound - lbound
    UL1bound(k) = UL1(k+lbound); yaabound(k) = yaa(k+lbound);
end

dyaa = (max(yaabound)-min(yaabound))/length(yaabound);
y = (-Ny/2:Ny/2-1)*lambda*fF/(Ny*dyaa);
UL = fft(UL1bound,Ny);

%modulation by SN0 to add contributions of other lenslets
yTrans = y+fF*sin(alpha); %translation on y-axis. SN0 is centered about the
coordinate y=-fF*sin(alpha)
SN0=sin(N0*pi*pL*yTrans./(lambda*fF))./sin(pi*pL*yTrans./(lambda*fF));
%modulation

```

```

SN0(isnan(SN0))=max(SN0); %remove the NaN point in the center

Isample = abs(UL.*SN0).^2; %return the intensity distribution to intalpha
end

function alphaResult = intalpha(lambda)
%define globals
global Nalpha alphaMaxRad alpha

%start calculations
for c = 1:Nalpha
    alpha = -alphaMaxRad+2*c*alphaMaxRad/(Nalpha-1)-2*alphaMaxRad/(Nalpha-1);
    alphaSum(c,:) = integrand(alpha,lambda); %perform calculations for this
alpha
end

alphaResult = sum(alphaSum,1)./Nalpha;
end

```