lin_global_search_faster (Calls: 1, Time: 1593.342 s)

Generated 29-Dec-2022 17:45:39 using performance time.
Script in file C:\Users\zacky\OneDrive\Documents\MATLAB\Membranes_Blood\lin_global_search_faster.m
Copy to new window for comparing multiple runs

Parents (calling functions)

No parent

Lines that take the most time

Line Number	Code	Calls	Total Time (s)	% Time	Time Plot
93	`[out,fval2,exitflag,output,solutions] = run(gs,proble…`	1	866.318	54.4%	▬▬▬
239	`alpha_B_test(pp,hh,aa) = fzero(@(x) lipid_con_bend_te…`	15625	709.318	44.5%	▬▬
241	`[epsilon, n0, d, R, kD, kappa, alpha_i, N]);`	15625	12.302	0.8%	
40	`@(y) lipid_con_phi(y,const), options);`	1	4.908	0.3%	
262	`squeeze(E_nans(min_phi:max_phi,min_h:max_h,ii))');`	25	0.108	0.0%	
All other lines			0.388	0.0%	
Totals			1593.342	100%	

Children (called functions)

Function Name	Function Type	Calls	Total Time (s)	% Time	Time Plot
MultiStart>MultiStart.run	Class method	1	866.318	54.4%	▬▬▬
fzero	Function	15625	709.088	44.5%	▬▬
lin_global_search_faster>stretch_bend_min	Subfunction	15625	12.235	0.8%	
fmincon	Function	1	4.906	0.3%	
surf	Function	25	0.106	0.0%	
hold	Function	4	0.090	0.0%	
optimoptions	Function	1	0.013	0.0%	
xlabel	Function	3	0.012	0.0%	
optimset	Function	1	0.010	0.0%	
createOptimProblem	Function	1	0.009	0.0%	
free_shape_linear_fixed_h	Function	1	0.009	0.0%	
rng	Function	1	0.006	0.0%	
newplotwrapper	Function	3	0.006	0.0%	
ylim	Function	3	0.006	0.0%	
ylabel	Function	3	0.005	0.0%	
RandomStartPointSet>RandomStartPointSet.RandomStartPointSet	Class method	1	0.005	0.0%	
MultiStart>MultiStart.MultiStart	Class method	1	0.004	0.0%	
linspace	Function	4	0.003	0.0%	
xlim	Function	1	0.001	0.0%	
CustomStartPointSet>CustomStartPointSet.CustomStartPointSet	Class method	1	0.001	0.0%	
lin_global_search_faster>@(x)x.Fval	Anonymous function	486	0.001	0.0%	
rad2deg	Function	26	0.000	0.0%	
squeeze	Function	25	0.000	0.0%	
deg2rad	Function	2	0.000	0.0%	
trapz	Function	1	0.000	0.0%	
Self time (built-ins, overhead, etc.)			0.507	0.0%	
Totals			1593.342	100%	

Code Analyzer results

Line Number	Message
82	`Variable appears to change size on every loop iteration (within a script). Consider preallocating for s…`
128	`Variable appears to change size on every loop iteration (within a script). Consider preallocating for s…`
129	`Variable appears to change size on every loop iteration (within a script). Consider preallocating for s…`
130	`Variable appears to change size on every loop iteration (within a script). Consider preallocating for s…`

131	Variable appears to change size on every loop iteration (within a script). Consider preallocating for s…
132	Variable appears to change size on every loop iteration (within a script). Consider preallocating for s…
133	Variable appears to change size on every loop iteration (within a script). Consider preallocating for s…
135	Variable appears to change size on every loop iteration (within a script). Consider preallocating for s…
136	Variable appears to change size on every loop iteration (within a script). Consider preallocating for s…
137	Variable appears to change size on every loop iteration (within a script). Consider preallocating for s…
138	Variable appears to change size on every loop iteration (within a script). Consider preallocating for s…
139	Variable appears to change size on every loop iteration (within a script). Consider preallocating for s…
140	Variable appears to change size on every loop iteration (within a script). Consider preallocating for s…
141	Variable appears to change size on every loop iteration (within a script). Consider preallocating for s…
145	This statement (and possibly following ones) cannot be reached.
199	Variable appears to change size on every loop iteration (within a script). Consider preallocating for s…
200	Variable appears to change size on every loop iteration (within a script). Consider preallocating for s…
201	Variable appears to change size on every loop iteration (within a script). Consider preallocating for s…
202	Variable appears to change size on every loop iteration (within a script). Consider preallocating for s…
203	Variable appears to change size on every loop iteration (within a script). Consider preallocating for s…
232	Add a semicolon after the statement to hide the output (in a script).

Coverage results

Show coverage for parent folder

Total lines in function	426
Non-code lines (comments, blank lines)	137
Code lines (lines that can run)	289
Code lines that did run	170
Code lines that did not run	119
Coverage (did run/can run)	58.82 %

Function listing

```
 Time     Calls    Line
  0.003        1      1  clear variables
                      2
                      3  % constants
 < 0.001       1      4  R = 0.05;                  % um
 < 0.001       1      5  sigma = 0.01;          % surface fraction
 < 0.001       1      6  d = sqrt(R^2/sigma);     % um
                      7  % phi = pi/12;
 < 0.001       1      8  kD  = 300/10^12*1e9;     % picoJ/um^2
 < 0.001       1      9  zeta = 0.02;             % dimensionless
 < 0.001       1     10  epsilon = -zeta*kD;      % picoJ/um^2
                     11  % epsilon = -1;
 < 0.001       1     12  n0 = 1;                  % fraction
 < 0.001       1     13  kappa = 1e-19*1e12;      % picoJ
                     14  % kappa = 1e-17*1e12;      % picoJ
 < 0.001       1     15  alpha i = 0.01;         % fraction
 < 0.001       1     16  zeta = epsilon*n0/kD;    % dimensionless
 < 0.001       1     17  N = 3e3;                 % number of points in quadrature
 < 0.001       1     18  plotfigs = 0;
                     19
                     20  % param 1 vals = logspace(-21,-15, 60)*1e12;
                     21  % param 1 vals = logspace(-4,0, 40);
                     22  % param 1 vals = logspace(-3,-1, 1);
 < 0.001       1     23  param 1 vals = 0.02;
                     24
 < 0.001       1     25  for ii = 1:length(param 1 vals)
                     26  %     kappa = param 1 vals(ii);
                     27  %     zeta = param 1 vals(ii);
                     28  %     epsilon = -zeta*kD;      % picoJ/um^2
                     29  %     kD = param 1 vals(ii);
                     30  %     kappa = 3.333e-7*kD;
 < 0.001       1     31      sigma = param 1 vals(ii);
 < 0.001       1     32      d = sqrt(R^2/sigma);     % um
                     33
                     34      % solve microplastics for initial stretch of A and B
  0.011      1       35      options = optimset('MaxFunEvals', 1e5, 'MaxIter', 1e4);
< 0.001      1       36      const = [zeta, alpha i, d, R];
  4.908      1       37      [out,fval,~,~,lam vals,grad,hessian] = ...
             1       38          fmincon(@(y) free phi(y, const),[alpha i, alpha i, 0.3], [],[],[],[],...
             1       39          [-1,-1,0],[Inf, Inf, pi/2], ...
             1       40          @(y) lipid con phi(y,const), options);
< 0.001      1       41      alpha A init = out(1);
< 0.001      1       42      alpha B init = out(2);
< 0.001      1       43      phi init = out(3);
< 0.001      1       44      h phi init = 0;
< 0.001      1       45      S A init = 2*pi*R^2*(1-cos(phi init));
< 0.001      1       46      S B init = d^2 - pi*R^2*sin(phi init)^2;
                     47
< 0.001      1       48      E adhesion init = epsilon*n0*S A init ./(1+alpha A init );
< 0.001      1       49      E stretch A init  = kD/2*(alpha A init .^2*S A init ./(1+alpha A init ));
< 0.001      1       50      E stretch B init  = kD/2*(alpha B init .^2*S B init ./(1+alpha B init ));
< 0.001      1       51      E total init = E adhesion init+E stretch A init+E stretch B init;
                     52
                     53      % use initial stretch to solve for minimum attachment
                     54
< 0.001      1       55      const = [epsilon, n0, d, R, kD, kappa, alpha i, N];
  0.013      1       56      opts = optimoptions(@fmincon,Algorithm="sqp");
  0.007      1       57      rng default % For reproducibility
  0.010      1       58      problem = createOptimProblem("fmincon",...
             1       59          x0 = [alpha A init, alpha B init, phi init, h phi init],...
             1       60          objective = @(y) stretch bend min(y, const),...
             1       61          lb = [-0.1,-0.1,0,-2*d],...
             1       62          ub = [0.1, 0.1, pi/2, 2*d],...
             1       63          nonlcon = @(y) lipid con bend(y,const),...
             1       64          options=opts);
                     65      % gs = GlobalSearch;
< 0.001      1       66      st = alpha i*2;
                     67  %     ptmatrix = [alpha A init, alpha B init, phi init, h phi init;...
                     68  %         -st, -st, 5, 0;-st, st, 5, 0;st, -st, 5, 0;st, st, 5, 0;...
                     69  %         -st, -st, 5, R;-st, -st, 5, -R;st, st, 5, d/2;st, st, 5, -d/2;...
                     70  %         -st, -st, 85, 0;-st, st, 85, 0;st, -st, 85, 0;st, st, 85, 0;...
                     71  %         -st, -st, 85, R;-st, -st, 85, -R;st, st, 85, d/2;st, st, 85, -d/2;...
                     72  %         -st, -st, 45, 0;-st, st, 45, 0;st, -st, 45, 0;st, st, 45, 0;...
                     73  %         -st, -st, 45, R;-st, -st, 45, -R;st, st, 45, d/2;st, st, 45, -d/2];
< 0.001      1       74      alpha A vals inp = [-0.1, -alpha i, -1e-3, 0, 1e-3, alpha i, 0.1];
< 0.001      1       75      alpha B vals inp = [-0.1, -alpha i, -1e-3, 0, 1e-3, alpha i, 0.1];
< 0.001      1       76      phi vals inp = deg2rad([0, 5, 20, 45, 60, 85, 90]);
< 0.001      1       77      h phi vals inp = [-2*d,-d/2, -R, 0, R, d/2,2*d];
< 0.001      1       78      for aa = 1:length(alpha A vals inp)
< 0.001      7       79          for bb = 1:length(alpha B vals inp)
< 0.001     49       80              for pp = 1:length(phi vals inp)
< 0.001    343       81                  for hh = 1:length(h phi vals inp)
  0.003   2401       82                      ptmatrix(aa,bb,pp,hh,1:4) = ...
          2401       83                          [alpha A vals inp(aa), alpha A vals inp(bb),...
          2401       84                           phi vals inp(pp), h phi vals inp(hh)];
< 0.001   2401       85                  end
< 0.001    343       86              end
< 0.001     49       87          end
< 0.001      7       88      end
< 0.001      1       89      ptmatrix = reshape(ptmatrix, [numel(ptmatrix)/4, 4]);
  0.002      1       90      tpoints = CustomStartPointSet(ptmatrix);
  0.005      1       91      rs = RandomStartPointSet('NumStartPoints',100);
  0.004      1       92      gs = MultiStart("FunctionTolerance",1e-3, "XTolerance", 1e-3);
866.318      1       93      [out,fval2,exitflag,output,solutions] = run(gs,problem, {tpoints,rs});
                     94  %     [out,fval2,exitflag,output,solutions] = run(gs,problem, {rs});
                     95  %     [out,fval2,exitflag,output,solutions] = run(gs,problem, {tpoints});
                     96
                     97  %     figure();
                     98  %     hold on
                     99  %     plot(arrayfun(@(x)x.Fval,solutions),'k*')
                    100  %     xlabel('Solution number')
                    101  %     ylabel('Function value')
                    102  %     title('Solution Function Values')
                    103
                    104      % get energies etc
```

```
< 0.001   1  105      alpha A = out(1);
< 0.001   1  106      alpha B = out(2);
< 0.001   1  107      phi = out(3);
< 0.001   1  108      h phi = out(4);
             109
             110      % get the shape of the free region
< 0.001   1  111      Sigma = kD*alpha B;
< 0.001   1  112      lambda = sqrt(kappa/Sigma);
< 0.001   1  113      r phi = sin(phi)*R;
  0.001   1  114      r = linspace(r phi, d/2,N);
             115
  0.009   1  116      [h,C,S B, ~, lap h, hderiv] = free shape linear fixed h(r, r phi, d, phi, kappa, Sigma, h phi);
             117
< 0.001   1  118      S A = 2*pi*R^2*(1-cos(phi));
             119
< 0.001   1  120      E adhesion = epsilon*n0*S A./(1+alpha A);
< 0.001   1  121      E stretch A = kD/2*(alpha A.^2*S A./(1+alpha A));
< 0.001   1  122      E stretch B = kD/2*(alpha B.^2*S B./(1+alpha B));
< 0.001   1  123      E bend B = kappa/2*2*pi*trapz(r, r.*lap h.^2);
< 0.001   1  124      E bend A = 4*pi*kappa*(1-cos(phi));
             125
< 0.001   1  126      E = E adhesion + E stretch A + E stretch B + E bend A + E bend B;
             127
< 0.001   1  128      E all(1,ii) = E;
< 0.001   1  129      E all(2,ii) = E adhesion;
< 0.001   1  130      E all(3,ii) = E stretch A;
< 0.001   1  131      E all(4,ii) = E stretch B;
< 0.001   1  132      E all(5,ii) = E bend A;
< 0.001   1  133      E all(6,ii) = E bend B;
             134
< 0.001   1  135      alpha A vals(ii) = alpha A;
< 0.001   1  136      alpha B vals(ii) = alpha B;
< 0.001   1  137      phi vals(ii) = phi;
< 0.001   1  138      h phi vals(ii) = h phi;
< 0.001   1  139      S A vals(ii) = S A;
< 0.001   1  140      S B vals(ii) = S B;
< 0.001   1  141      Sigma vals(ii) = Sigma;
             142
             143      % plot of shape and nanoparticle
< 0.001   1  144      if plotfigs
             145          figure('Position',[400,100,700,500]);
             146          hold on
             147          axis equal
             148          xlabel('$r$')
             149          ylabel('$h$')
             150          plot(r, h, 'displayname', 'free surface');
             151          t = linspace(-pi/2,pi/2,1000);
             152          x = cos(t)*R;
             153          % y = sin(t)*R+(R*cos(phi)+h(1));
             154          y = sin(t)*R+R*cos(phi)+h(1);
             155          plot(x,y, 'displayname', 'microbead')
             156          % plot(x, (x-sin(phi))*tan(phi)+h(1))
             157          % legend('location', 'se')
             158          annotation('textbox', [0.3,0.7,0.4,0.2], 'String',...
             159              [sprintf('$R = %0.2g$ $\\mu$m \n', R),...
             160              sprintf('$k D = %0.2g$ pJ/$\\mu$m$^2$ \n', kD),...
             161              sprintf('$\\alpha i = %0.2g$ \n', alpha i),...
             162              sprintf('$d^2 = %0.2g R^2$ \n', sigma),...
             163              sprintf('$\\epsilon n 0 = %0.2g k \\mathrm{D}$ \n', zeta),...
             164              sprintf('$\\kappa = %0.2g k \\mathrm{D} R^2$ \n', kappa/(kD*R^2))], ...
             165              'interpreter', 'latex','FontSize',16,...
             166              'FitBoxToText','on','LineStyle','none', 'Color','b')
             167          annotation('textbox', [0.6,0.7,0.4,0.2], 'String',...
             168              [sprintf('$\\alpha A = %0.2g$ \n', alpha A),...
             169              sprintf('$\\alpha B = %0.2g$ \n', alpha B),...
             170              sprintf('$\\phi = %0.2g^{\\circ}$ \n', rad2deg(phi)),...
             171              sprintf('$h \\phi = %0.2g$ $\\mu$m \n', h phi)], ...
             172              'interpreter', 'latex','FontSize',16,...
             173              'FitBoxToText','on','LineStyle','none', 'Color','k')
             174          annotation('textbox', [0.55,0.2,0.4,0.2], 'String',...
             175              [sprintf('$E = %0.3g$ pJ \n', E)], ...
```

```
                 176              'interpreter', 'latex','FontSize',16,...
                 177              'FitBoxToText','on','LineStyle','none', 'Color','r')
< 0.001        1 178      end
                 179
< 0.001        1 180  end
                 181
                 182  % save('kappa sweep R300sig01kD300zeta02ai01.mat')
                 183
                 184  %% plotting solution space
  0.029        1 185  figure();
  0.062        1 186  hold on
  0.006        1 187  ylim([min(cat(1,solutions.Fval))*1.1, 1e-5])
  0.019        1 188  plot(arrayfun(@(x)x.Fval,solutions),'k*')
  0.010        1 189  xlabel('Solution number')
  0.003        1 190  ylabel('Function value')
                 191  % title('Solution Function Values')
                 192
  0.033        1 193  figure();
  0.008        1 194  hold on
                 195  % zlim([-1e-3, 1e-5])
< 0.001        1 196  for ii = 1:size(solutions,2)
  0.001      486 197      X0 = solutions(ii).X0{1};
  0.001      486 198      X = solutions(ii).X;
< 0.001      486 199      phi val(ii) = X(3);
< 0.001      486 200      h phi val(ii) = X(4);
  0.002      486 201      E val(ii) = solutions(ii).Fval;
< 0.001      486 202      dist(ii) = vecnorm(out-X);
< 0.001      486 203      E dist(ii) = E val(ii) - fval2;
                 204  %     plot3(rad2deg(phi val(ii)), h phi val(ii), E val(ii), 'k.')
< 0.001      486 205  end
  0.002        1 206  xlim([0,90])
< 0.001        1 207  ylim([min(cat(1,solutions.Fval))*1.1, 1e-5])
  0.002        1 208  plot(rad2deg(phi val), E val, 'o');
                 209  % surf(phi val, h phi val, E val);
                 210  % plot(arrayfun(@(x)x.Fval,solutions),'k*')
  0.001        1 211  xlabel('$\phi$')
  0.001        1 212  ylabel('$E$')
                 213  % title('Solution Function Values')
                 214
  0.028        1 215  figure();
  0.008        1 216  hold on
< 0.001        1 217  ylim([0, 1e-4])
< 0.001        1 218  plot(dist, E dist, '*')
  0.001        1 219  xlabel('$\Delta X$')
< 0.001        1 220  ylabel('$\Delta E$')
                 221
                 222  %% get surfaces
< 0.001        1 223  size phi = 25;
< 0.001        1 224  size h = 25;
< 0.001        1 225  size A = 25;
< 0.001        1 226  phi test vals = deg2rad(linspace(eps(1),90-eps(1),size phi));
  0.002        1 227  h phi test vals = linspace(-2*R, 2*R, size h);
< 0.001        1 228  alpha A test vals = linspace(-0.1,0.1,size A);
< 0.001        1 229  alpha B test = zeros(size phi,size h,size A);
< 0.001        1 230  E test = zeros(size phi,size h,size A);
< 0.001        1 231  for aa = 1:length(alpha A test vals)
  0.007       25 232      alpha A test = alpha A test vals(aa)
< 0.001       25 233      for pp = 1:length(phi test vals)
< 0.001      625 234          phi test = phi test vals(pp);
< 0.001      625 235          for hh = 1:length(h phi test vals)
  0.006    15625 236              h phi test = h phi test vals(hh);
  0.014    15625 237              const = [epsilon, n0, d, R, kD, kappa, alpha i, N,...
           15625 238                  alpha A test, phi test, h phi test];
709.318    15625 239              alpha B test(pp,hh,aa) = fzero(@(x) lipid con bend test(x, const), rand()*0.2-0.1);
 12.312    15625 240              E test(pp,hh,aa) = stretch bend min([alpha A test, alpha B test(pp,hh,aa), phi test, h phi test],
           15625 241                  [epsilon, n0, d, R, kD, kappa, alpha i, N]);
  0.005    15625 242          end
< 0.001      625 243      end
< 0.001       25 244  end
                 245
                 246  %%
```

```
< 0.001     1  247  min phi = 1;
< 0.001     1  248  max phi = 25;
< 0.001     1  249  min h = 1;
< 0.001     1  250  max h = 25;
< 0.001     1  251  E nans = E test;
< 0.001     1  252  E nans(alpha B test<-0.1|alpha B test>0.1) = NaN;
  0.030     1  253  figure();
               254  % zlim([-1,1])
  0.013     1  255  hold on
< 0.001     1  256  for ii=1:size A
               257  % surf(rad2deg(phi test vals(min phi:max phi)),...
               258  %     h phi test vals(min h:max h),...
               259  %     squeeze(E test(min phi:max phi,min h:max h,1))');
  0.109    25  260  surf(rad2deg(phi test vals(min phi:max phi)),...
           25  261      h phi test vals(min h:max h),...
           25  262      squeeze(E nans(min phi:max phi,min h:max h,ii))');
< 0.001    25  263  end
               264
               265  % squeeze(alpha B test(min phi:max phi,min h:max h,1))
               266
< 0.001     1  267  [min val, Q] = min(E nans, [], 'all');
< 0.001     1  268  sE = size(E nans);
< 0.001     1  269  ct = floor(Q/(sE(1)*sE(2)))+1;
< 0.001     1  270  bt = floor((Q - (ct-1)*(sE(1)*sE(2)))/sE(1))+1;
< 0.001     1  271  at = floor(Q-(ct-1)*(sE(1)*sE(2))-(bt-1)*sE(1));
               272
               273  % phi min = rad2deg(phi test vals(at));
< 0.001     1  274  phi min = phi test vals(at);
< 0.001     1  275  h phi min = h phi test vals(bt);
< 0.001     1  276  alpha A min = alpha A test vals(ct);
< 0.001     1  277  alpha B min = alpha B test(at,bt,ct);
< 0.001     1  278  out approx min = [alpha A min, alpha B min, phi min, h phi min];
               279
               280  %% plotting
< 0.001     1  281  init vals = param 1 vals(1:20);
               282  on = ones(size(init vals));
               283
               284  figure('Position',[400,100,800,600]);
               285  hold on
               286  axes1 = gca;
               287  axes1.XScale = 'log';
               288  % xlabel('$\kappa$ (pJ)')
               289  % xlabel('$\zeta$')
               290  xlabel('surface fraction $\sigma$')
               291  ylabel('E (pJ)')
               292  lines = ["-", ":", ":", "--", ":", "--"];
               293  colours = ['k', 'b', 'r', 'r', 'g', 'g'];
               294  for ii=1:6
               295      plot(param 1 vals, E all(ii,:), ...
               296          strcat(colours(ii),lines(ii)))
               297  end
               298  % plot(init vals, on*E total init, 'k-.', 'linewidth', 0.5);
               299  % plot(init vals, on*E adhesion init, 'b-.', 'linewidth', 0.5);
               300  % plot(init vals, on*E stretch B init, 'r-.', 'linewidth', 0.5);
               301  legend({'$E \mathrm{total}$','$E \mathrm{adhesion}$',...
               302      '$E \mathrm{stretch,A}$','$E \mathrm{stretch,B}$',...
               303      '$E \mathrm{bend,A}$','$E \mathrm{bend,B}$'}, 'Box','off',...
               304      'location', 'best')
               305  annotation('textbox', [0.3,0.6,0.4,0.2], 'String',...
               306      [sprintf('$R = %0.2g$ $\\mu$m \n', R),...
               307      sprintf('$k D = %0.2g$ pJ/$\\mu$m$^2$ \n', kD),...
               308      sprintf('$\\alpha i = %0.2g$ \n', alpha i),...
               309      sprintf('$d^2 = %0.2g R^2$ \n', sigma),...
               310      sprintf('$\\epsilon n 0 = %0.2g k \\mathrm{D}$ \n', zeta)],...
               311      'interpreter', 'latex','FontSize',16,...
               312      'FitBoxToText','on','LineStyle','none', 'Color','b')
               313
               314  % positive percentages
               315  figure('Position',[400,100,800,600]);
               316  hold on
               317  axes1 = gca;
```

```
axes1.XScale = 'log';
% xlabel('$\kappa$ (pJ)')
% xlabel('$\zeta$')
xlabel('surface fraction $\sigma$')
ylabel('Fraction of Energy')
lines = ["-", ":", ":", "--", ":", "--"];
colours = ['k', 'b', 'r', 'r', 'g', 'g'];
E pos = E all(1,:)-E all(2,:);
E pos 2 = sum(E all(3:end,:));
E perc = E all./E pos;
for ii=3:6
    plot(param 1 vals, E perc(ii,:), ...
        strcat(colours(ii),lines(ii)))
end
legend({'$E \mathrm{stretch,A}$','$E \mathrm{stretch,B}$',...
    '$E \mathrm{bend,A}$','$E \mathrm{bend,B}$'}, 'Box','off',...
    'location', 'best')

% areas
figure('Position',[400,100,800,600]);
hold on
axes1 = gca;
axes1.XScale = 'log';
% xlim([0,90]);
% xlabel('$\kappa$ (pJ)')
% xlabel('$\zeta$')
xlabel('surface fraction $\sigma$')
yyaxis left
ylabel('Area')
% plot(init vals, on*S B init, '-.', 'linewidth', 0.5, 'HandleVisibility','off');
% plot(init vals, on*(S B init+S A init), '-.', 'linewidth', 0.5, 'HandleVisibility','off');
plot(param 1 vals, S B vals, '--', 'displayname', '$S B$')
plot(param 1 vals, S A vals+S B vals, '-', 'displayname', '$S \mathrm{total}$')
yyaxis right
ylabel('Area')
% plot(init vals, on*S A init, '-.', 'linewidth', 0.5, 'HandleVisibility','off');
plot(param 1 vals, S A vals, '--', 'displayname', '$S A$')
legend

% stretches
figure('Position',[400,100,800,600]);
hold on
axes1 = gca;
axes1.XScale = 'log';
% xlim([0,90]);
% xlabel('$\kappa$ (pJ)')
% xlabel('$\zeta$')
xlabel('surface fraction $\sigma$')
ylabel('$\alpha$')
plot(param 1 vals, alpha B vals, 'displayname', '$\alpha B$')
plot(param 1 vals, alpha A vals, 'displayname', '$\alpha A$')
% plot(init vals, on*alpha A init, 'r-.', 'linewidth', 0.5, 'HandleVisibility','off');
% plot(init vals, on*alpha B init, 'b-.', 'linewidth', 0.5, 'HandleVisibility','off');
legend

% phi and h phi
figure('Position',[400,100,800,600]);
hold on
axes1 = gca;
axes1.XScale = 'log';
% xlim([0,90]);
% xlabel('$\kappa$ (pJ)')
% xlabel('$\zeta$')
xlabel('surface fraction $\sigma$')
yyaxis left
ylabel('$\phi$')
plot(param 1 vals, rad2deg(phi vals), 'displayname', '$\phi$')
% plot(init vals, on*rad2deg(phi init), '-.', 'linewidth', 0.5, 'HandleVisibility','off')
yyaxis right
ylabel('$h \phi$')
plot(param 1 vals, h phi vals, 'displayname', '$h \phi$')
```

```
legend

%% replotting each function

figure('Position',[400,100,700,500]);
hold on
axis equal
xlabel('$r$')
ylabel('$h$')
for ii = [1,5,10,20,30,40]
    sigma = param 1 vals(ii);
    d = sqrt(R^2/sigma);    % um

    alpha A = alpha A vals(ii);
    alpha B = alpha B vals(ii);
    phi = phi vals(ii);
    h phi = h phi vals(ii);

    % get the shape of the free region
    Sigma = kD*alpha B;
    lambda = sqrt(kappa/Sigma);
    r phi = sin(phi)*R;
    r = linspace(r phi, d/2,N);

    [h,C,S B, ~, lap h, hderiv] = free shape linear fixed h(r, r phi, d, phi, kappa, Sigma, h phi);

    h1 = plot(r, h, 'displayname', sprintf('$\\kappa = %0.2e$', kappa));
    colour = h1.Color;
    t = linspace(-pi/2,-pi/2+phi,1000);
    x = cos(t)*R;
    % y = sin(t)*R+(R*cos(phi)+h(1));
    y = sin(t)*R+R*cos(phi)+h(1);
    plot(x,y, ':', 'HandleVisibility','off', 'color', colour)
end

legend

%% extra functions
```

Local functions in this file are not included in this listing.