```matlab
function [ q, b1, s, b3, h ] = bsbfun(z0, th0, zf, thf, e1, e3, ths,
 varargin)
% bsbfun -- BSB computation and plot.
%
%  Usage
%    [ b1, s, b3, q, h ] = bsbfun(z0, th0, zf, thf, e1, e3, ths, symbol)
%
%  Inputs
%    z0     complex, initial condition
%    th0    real, initial condition
%    zf     complex, terminal condition
%    thf    real, terminal condition
%    e1     integer, +1/-1
%    e3     integer, +1/-1
%    ths    real, singular value of angle
%    symbol character, used for plot
%    N      integer, number of points for plot [ 100 ]
%
%  Outputs
%    q      complex, product of z(tf-b3)-z(b1) with conjugate of
 exp(i.ths)+w
%    b1     real, duration on the first bang arc
%    s      real, duration on the singular arc
%    b3     real, duration on the last bang arc
%    h      integer, handle to the current plot
%
%  Description
%    Computes a BSB sequence. The sequence is admissible provided Re q > 0
 and Im q = 0.
%    Plot if a symbol is passed.
%

global w

if (nargin == 7)
  draw = 0;
elseif (nargin == 8)
  draw = 1;
  bcol = [ 'k' varargin{1} ];
  scol = [ 'r' varargin{1} ];
  N = 100;
elseif (nargin == 9)
  draw = 1;
  bcol = [ 'k' varargin{1} ];
  scol = [ 'r' varargin{1} ];
  N = varargin{2};
else
  error('Bad number of input arguments.')
end;

th0 = angle(exp(1i*th0)); % normalization to (-pi,pi]
thf = angle(exp(1i*thf));
ths = angle(exp(1i*ths));
                                        3.1
b1 = ( ths - th0 ) / e1;

if b1 < 0, b1 = b1+2*pi; end;
if b1 < 0, error('Bad b1.'); end;
                                               3.1
b3 = ( thf - ths ) / e3;
if b3 < 0, b3 = b3+2*pi; end;
if b3 < 0, error('Bad b3.'); end;

ih = ishold;

if draw, t = linspace(0, b1, N); else t = b1; end;
th = th0 + e1*t;
z = -1i*e1*( exp(1i*th) - exp(1i*th0) ) + w*t + z0;
if draw
  plot(z, bcol), hold on;
  quiver(real(z0), imag(z0), real(exp(1i*th0)+w), imag(exp(1i*th0)+w));
end;
z1 = z(end);

if draw, t = linspace(-b3, 0, N); else t = -b3; end;
th = thf + e3*t;
z = -1i*e3*( exp(1i*th) - exp(1i*thf) ) + w*t + zf;
if draw
  plot(z, bcol);
  quiver(real(zf), imag(zf), real(exp(1i*thf)+w), imag(exp(1i*thf)+w));
end;
z2 = z(1);

if draw
  plot(linspace(real(z1), real(z2), N), linspace(imag(z1), imag(z2), N),
   scol);
end;

s = abs(z2-z1) / abs(exp(1i*ths)+w);

q = (z2-z1) * (exp(1i*ths)+w)';

if ~ih, hold off; end

if draw, h = gcf; else h = 0; end;

% Written on Mon  5 Nov 2018 18:26:31 CET
% by Jean-Baptiste Caillau - Universite Cote d'Azur, CNRS, Inria, LJAD
```

```
function [ llu1, llu3, llu5, winner ] = game(lu1, lu3, lu5, inter, dsp)
% game -- Hexapawn game
%
%  Usage
%    [ llu1, llu3, llu5, winner ] = game(lu1, lu3, lu5)
%    [ llu1, llu3, llu5, winner ] = game(lu1, lu3, lu5, inter, dsp)
%
%  Inputs
%    lu1    list, possible controls for u1 depending on X1
%    lu3    list, possible controls for u3 depending on X3
%    lu5    list, possible controls for u5 depending on X5
%    inter  boolean, interactive player1 [ False ]
%    dsp    boolean, display states [ False ]
%
%  Outputs
%    llu1   list, reinforcement of lu1
%    llu3   list, reinforcement of lu3
%    llu5   list, reinforcement of lu5
%    winner integer, 1 or 2
%
%  Description
%    Plays one Hexapawn game and reinforces player2 controls.
%
%  See also
%    reinforce
%

global lX1
global lX3
global lX5

if (nargin == 3)
  inter = 0;
  dsp = 0;
elseif (nargin == 4)
  dsp = 0;
end;

X0 = [ 2 2 2
       0 0 0
     1 1 1 ];
if dsp, disp('Initial game:'); disp(X0); end;

% Move 0: player1
while 1
  u0 = play1(X0, inter);
  if norm(u0 - [ 3 3 ; 2 3 ]) > 0,
    break;
  else
    if inter, disp('No right opening!'); end;
  end;
end;

X1 = f1(X0, u0); % no possible win after u0
if dsp, disp('Player 1 move:'); disp(X1); end;

% Move 1: player2
u1 = play2(X1, lX1, lu1);
X2 = f2(X1, u1); % no possible win after u1
if dsp, disp('Player 2 move:'); disp(X2); end;

% Move 2: player1
u2 = play1(X2, inter);
X3 = f1(X2, u2);
if dsp, disp('Player 1 move:'); disp(X3); end;

if win1(X3) | isempty(play2(X3, lX3, lu3))        3.3
  winner = 1;
  lu1 = reinforce(X1, u1, lX1, lu1);

else

% Move 3: player2
u3 = play2(X3, lX3, lu3);
X4 = f2(X3, u3);
if dsp, disp('Player 2 move:'); disp(X4); end;

if win2(X4)
  winner = 2;

else

% Move 4: player1
u4 = play1(X4, inter);
X5 = f1(X4, u4);
if dsp, disp('Player 1 move:'); disp(X5); end;

if win1(X5) | isempty(play2(X5, lX5, lu5))        3.3
  winner = 1;
  lu3 = reinforce(X3, u3, lX3, lu3);

else

% Move 5: player2
u5 = play2(X5, lX5, lu5);
X6 = f2(X5, u5);
if dsp, disp('Player 2 move:'); disp(X6); end;

if win2(X6)
  winner = 2;

else
  winner = 1; % useless to play last move
  lu5 = reinforce(X5, u5, lX5, lu5);

end, end, end, end;

llu1 = lu1;
llu3 = lu3;
llu5 = lu5;
```