```
%% Copyright (C) 2005 Julius O. Smith III
%%
%% This program is free software; you can redistribute it and/or modify it
%% under the terms of the GNU General Public License as published by
%% the Free Software Foundation; either version 2, or (at your option)
%% any later version.
%%
%% This program is distributed in the hope that it will be useful, but
%% WITHOUT ANY WARRANTY; without even the implied warranty of
%% MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  The GNU
%% General Public License has more details.
%%
%% You should have received a copy of the GNU General Public License
%% along with this program; see the file COPYING.  If not, see
%% <http://www.gnu.org/licenses/>.

%% -*- texinfo -*-
%% @deftypefn {Function File} {[@var{r}, @var{p}, @var{f}, @var{m}] =} residuez (@var{B}, @var{A})
%% Compute the partial fraction expansion of filter @math{H(z) = B(z)/A(z)}.
%%
%% INPUTS:
%% @var{B} and @var{A} are vectors specifying the digital filter @math{H(z) = B(z)/A(z)}.
%% Say @code{help filter} for documentation of the @var{B} and @var{A}
%% filter coefficients.
%%
%% RETURNED:
%%   @itemize
%%   @item @var{r} = column vector containing the filter-pole residues@*
%%   @item @var{p} = column vector containing the filter poles@*
%%   @item @var{f} = row vector containing the FIR part, if any@*
%%   @item @var{m} = column vector of pole multiplicities
%%   @end itemize
%%
%% EXAMPLES:
%% @example
%%   Say @code{test residuez verbose} to see a number of examples.
%% @end example
%%
%% For the theory of operation, see
%% @indicateurl{http://ccrma.stanford.edu/~jos/filters/residuez.html}
%%
%% @seealso{residue residued}
%% @end deftypefn

%%%%%%%%%%% old_code %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%function [r, p, f, m] = residuez(B, A, tol)
%%%%%%%%%%% new_code %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
           function [r, p, f, m] = bg_residuez(B, A, tol)
%%%%%%%%%%% end of new_code %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% RESIDUEZ - return residues, poles, and FIR part of B(z)/A(z)
%
% Let nb = length(b), na = length(a), and N=na-1 = no. of poles.
% If nb<na, then f will be empty, and the returned filter is
%
%             r(1)                      r(N)
% H(z) = ----------------  + ... + ----------------- = R(z)
%        [ 1-p(1)/z ]^m(1)         [ 1-p(N)/z ]^m(N)
%
% If, on the other hand, nb >= na, the FIR part f will not be empty.
% Let M = nb-na+1 = order of f = length(f)-1). Then the returned filter is
%
% H(z) = f(1) + f(2)/z + f(3)/z^2 + ... + f(M+1)/z^M + R(z)
%
% where R(z) is the parallel one-pole filter bank defined above.
% Note, in particular, that the impulse-response of the one-pole
% filter bank is in parallel with that of the the FIR part.  This can
% be wasteful when matching the initial impulse response is important,
% since F(z) can already match the first N terms of the impulse
% response. To obtain a decomposition in which the impulse response of
% the IIR part R(z) starts after that of the FIR part F(z), use RESIDUED.
%
% J.O. Smith, 9/19/05

if nargin==3
  warning("tolerance ignored");
end
NUM = B(:)'; DEN = A(:)';
% Matlab's residue does not return m (since it is implied by p):
%%%%%%%%%%% old_code %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%[r,p,f,m]=residue(conj(fliplr(NUM)),conj(fliplr(DEN)));
%%%%%%%%%%% new_code %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
                                        global  G_FID                  ;
                                        global  G_DEBUG                  ;
                                        if G_DEBUG
                                            fprintf(G_FID,'Converting a and b to A and B in bg_residuez gives\n');
```

```
                                        bg_print('B',conj(fliplr(NUM)));
                                        bg_print('A',conj(fliplr(DEN)));
                                        fprintf(G_FID,'---------------------------------------- ');
                                        fprintf(G_FID,'Into [R,P,F,M]=bg_residue(B,A,"MODE_Z");\n');
                                    end
[r,p,f,m]=bg_residue(conj(fliplr(NUM)),conj(fliplr(DEN)),"MODE_Z");
                                    if G_DEBUG
                                        fprintf(G_FID,'--------------------------------------- ');
                                        fprintf(G_FID,'Out of [R,P,F,M]=bg_residue(B,A,"MODE_Z");\n');
                                        fprintf(G_FID,'Outputs from bg_residue are:\n');
                                        bg_print('R',r);
                                        bg_print('P',p);
                                        bg_print('F',f);
                                        bg_print('M',m);
                                    end
%%%%%%%%%%% end of new_code %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
p = 1 ./ p;
r = r .* ((-p) .^m);
if f, f = conj(fliplr(f)); end

%%%%%%%%%%% new_code %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
                                    if G_DEBUG
                                        fprintf(G_FID,'Converting R,P,F and M to r,p,f and m ');
                                        fprintf(G_FID,'in bg_residuez gives;\n');
                                        bg_print('r',r);
                                        bg_print('p',p);
                                        bg_print('f',f);
                                        bg_print('m',m);
                                    end
endfunction
%%%%%%%%%%% end of new_code %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%!test
%! B=[1 -2 1]; A=[1 -1];
%%%%%%%%%%% old_code %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%                                                                %! [r,p,f,m] = residuez(B,A);
%%%%%%%%%%% new_code %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%! [r,p,f,m] = bg_residuez(B,A);
%%%%%%%%%%% end of new_code %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%! [r,p,f,m] = residuez(B,A);
%! assert(r,0,100*eps);
%! assert(p,1,100*eps);
%! assert(f,[1 -1],100*eps);
%! assert(m,1,100*eps);

%!test
%! B=1; A=[1 -1j];
%%%%%%%%%%% old_code %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%                                                                %! [r,p,f,m] = residuez(B,A);
%%%%%%%%%%% new_code %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%! [r,p,f,m] = bg_residuez(B,A);
%%%%%%%%%%% end of new_code %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%! assert(r,1,100*eps);
%! assert(p,1j,100*eps);
%! assert(f,[],100*eps);
%! assert(m,1,100*eps);

%!test
%! B=1; A=[1 -1 .25];
%%%%%%%%%%% old_code %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%                                                               %! [r,p,f,m] = residuez(B,A);
%%%%%%%%%%% new_code %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%! [r,p,f,m] = bg_residuez(B,A);
%%%%%%%%%%% end of new_code %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%! [rs,is] = sort(r);
%! assert(rs,[0;1],1e-7);
%! assert(p(is),[0.5;0.5],1e-8);
%! assert(f,[],100*eps);
%! assert(m(is),[1;2],100*eps);

%!test
%! B=1; A=[1 -0.75 .125];
%%%%%%%%%%% old_code %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%                                                               %! [r,p,f,m] = residuez(B,A);
%%%%%%%%%%% new_code %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%! [r,p,f,m] = bg_residuez(B,A);
%%%%%%%%%%% end of new_code %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%! [rs,is] = sort(r);
%! assert(rs,[-1;2],100*eps);
%! assert(p(is),[0.25;0.5],100*eps);
%! assert(f,[],100*eps);
%! assert(m(is),[1;1],100*eps);

%!test
%! B=[1,6,2]; A=[1,-2,1];
%%%%%%%%%%% old_code %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%                                                               %! [r,p,f,m] = residuez(B,A);
```

```
%%%%%%%%%%% new_code %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%! [r,p,f,m] = bg_residuez(B,A);
%%%%%%%%%%% end of new_code %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%! [rs,is] = sort(r);
%! assert(rs,[-10;9],1e-7);
%! assert(p(is),[1;1],1e-8);
%! assert(f,[2],100*eps);
%! assert(m(is),[1;2],100*eps);

%!test
%! B=[6,2]; A=[1,-2,1];
%%%%%%%%%%% old_code %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%                                                                 %! [r,p,f,m] = residuez(B,A);
%%%%%%%%%%% new_code %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%! [r,p,f,m] = bg_residuez(B,A);
%%%%%%%%%%% end of new_code %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%! [rs,is] = sort(r);
%! assert(rs,[-2;8],1e-7);
%! assert(p(is),[1;1],1e-8);
%! assert(f,[],100*eps);
%! assert(m(is),[1;2],100*eps);

%!test
%! B=[1,6,6,2]; A=[1,-2,1];
%%%%%%%%%%% old_code %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%                                                                 %! [r,p,f,m] = residuez(B,A);
%%%%%%%%%%% new_code %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%! [r,p,f,m] = bg_residuez(B,A);
%%%%%%%%%%% end of new_code %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%! [rs,is] = sort(r);
%! assert(rs,[-24;15],2e-7);
%! assert(p(is),[1;1],1e-8);
%! assert(f,[10,2],100*eps);
%! assert(m(is),[1;2],100*eps);

%!test
%! B=[1,6,6,2]; A=[1,-(2+j),(1+2j),-j];
%%%%%%%%%%% old_code %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%                                                                 %! [r,p,f,m] = residuez(B,A);
%%%%%%%%%%% new_code %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%! [r,p,f,m] = bg_residuez(B,A);
%%%%%%%%%%% end of new_code %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%! [rs,is] = sort(r);
%! assert(rs,[-2+2.5j;7.5+7.5j;-4.5-12j],1E-6);
%! assert(p(is),[1j;1;1],1E-6);
%! assert(f,-2j,1E-6);
%! assert(m(is),[1;2;1],1E-6);

%!test
%! B=[1,0,1]; A=[1,0,0,0,0,-1];
%%%%%%%%%%% old_code %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%                                                                 %! [r,p,f,m] = residuez(B,A);
%%%%%%%%%%% new_code %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%! [r,p,f,m] = bg_residuez(B,A);
%%%%%%%%%%% end of new_code %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%! [as,is] = sort(angle(p));
%! rise = [ ...
%!  0.26180339887499 - 0.19021130325903i; ...
%!  0.03819660112501 + 0.11755705045849i; ...
%!  0.4; ...
%!  0.03819660112501 - 0.11755705045849i; ...
%!  0.26180339887499 + 0.19021130325903i;];
%! pise = [ ...
%! -0.80901699437495 - 0.58778525229247i; ...
%!  0.30901699437495 - 0.95105651629515i; ...
%!  1; ...
%!  0.30901699437495 + 0.95105651629515i; ...
%! -0.80901699437495 + 0.58778525229247i];
%! assert(r(is),rise,100*eps);
%! assert(p(is),pise,100*eps);
%! assert(f,[],100*eps);
%! assert(m,[1;1;1;1;1],100*eps);
%%%%%%%%%%% new_code %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%!test
%! z1=+7.0372976777e+06;
%! p1=-3.1415926536e+09;
%! p2=-4.9964813512e+08;
%! r1=-(1+z1/p1)/(1-p1/p2)/p2/p1  ;
%! r2=-(1+z1/p2)/(1-p2/p1)/p2/p1  ;
%! r3=(1+(p2+p1)/p2/p1*z1)/p2/p1  ;
%! r4=z1/p2/p1                    ;
%! fi= 1.05e+08     ;
%! T =1/fi          ;
%! p1=exp(p1*T);
%! p2=exp(p2*T);
%! p3=1;
```

```
%! p4=1;
%! p =[p1;p2;p3;p4];
%! r4=r4*T;
%! r3=r3-r4;
%! r =[r1;r2;r3;r4];
%! m=[       1     ,     1     ,     1          ,     1        ;
%!        -(2+p2)  ,  -(2+p1)  , -(1+p1+p2)    ,  -(p1+p2)     ;
%!       +(1+2*p2) , +(1+2*p1) , (p1*p2+p1+p2) ,   +p1*p2      ;
%!          -p2    ,   -p1     , -p1*p2        ,     0         ];
%! a1=1.00;
%! a2=-(2+p1+p2);
%! a3=1+2*(p1+p2)+p1*p2;
%! a4=-(p1+p2+2*p1*p2);
%! a5=p2*p1;
%! a =[a1,a2,a3,a4,a5];
%! b=[m*r]';
%! k=[];
%! e=[1;1;1;2];
%! [rx,px,kx,ex] = bg_residuez(b,a);
%! assert ((abs (rx - r) < 1e-8
%!       && abs (px - p) < 1e-8
%!       && isempty (kx)
%!       && ex == e));
%%%%%%%%%%% end of new_code %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```