AAA Rational Approximation

Nick Trefethen, Oxford and KU Leuven

AAA algorithm

(= "adaptive Antoulas-Anderson")

THE AAA ALGORITHM FOR RATIONAL APPROXIMATION

YUJI NAKATSUKASA*, OLIVIER SÈTE†, AND LLOYD N. TREFETHEN‡

For Jean-Paul Berrut, the pioneer of numerical algorithms based on rational barycentric representations, on his 65th birthday.

SISC 2018

$$r(z) = \frac{n(z)}{d(z)} = \sum_{k=1}^{m} \frac{a_k}{z - z_k} \Big/ \sum_{k=1}^{m} \frac{b_k}{z - z_k}$$

- Fix $a_k = f_k b_k$, so that we are in interpolatory mode: $r(z_k) = f_k$.
- Taking $m = 1,2, \ldots$, choose support points z_m one after another.
- Next support point: sample point ζ_i where error $|f_i - r(\zeta_i)|$ is largest.
- Barycentric weights $\{b_k\}$ at each step:
  chosen to minimize linearized least-squares error $||\,fd - n\,||$.

AAA is remarkably effective, quickly producing approximations often within factor ~ 10 of optimal.
The support points tend to cluster near singularities, giving stability even in extreme cases.
AAA is least reliable on a real interval — because poles tend to appear in the interval.

Some AAA examples Nick Trefethen, KU Leuven, 6 May 2022

```
%% gammaplot.m - fplot of AAA approx of Gamma(x), based on 50 values in [-1,1]
X = linspace(-1,1,50)';
r = aaa(gamma(X),X);
clf, fplot(@gamma,[-3.5,4.5],'linewidth',1.5), hold on
fplot(r,[-3.5,4.5],'linewidth',1.5)
plot(X,0*X,'.k'), hold off
grid on, title('Gamma function based on 50 points in [-1,1]')
legend('Gamma(x)','AAA approx','location','southeast')

%% rounding.m - AAA approx of exp(x), showing local machine epsilons
X = linspace(-1,1,500)';
[r,poles] = aaa(exp(X),X);
degree = length(poles)
clf, plot(X,[0*X eps(exp(X))],'-r','linewidth',1.2), hold on
plot(X,exp(X)-r(X),'.k','markersize',7), grid on, hold off
ylim(2e-15*[-1 1])
title('Error in AAA approx of exp(x), and machine epsilon'), xlabel(' ')

%% branchcut.m - approximation of sqrt(z) and phase portrait of result
Z = rand(500,1) + 1i*rand(500,1);
[r,pol] = aaa(sqrt(Z),Z);
clf, plot(Z,'.k','markersize',5)
axis([-2 2 -2 2]), axis square, hold on
plot(pol,'.r','markersize',10)
title('sqrt(X) in 500 random pts in a square')
disp('type enter to see phase portrait'), pause
phaseplot(r)

%% beam.m - NICONET model order reduction beam example
clf, load beam                 % matrix A of dimension 348
I = eye(size(A));
f = @(z) C*((z*I-A)\B);        % scalarized transfer function
npts = 500;
Z0 = 1i*logspace(-2,2,npts);   % sample points on positive imaginary axis
fZ0 = [];
for ii = 1:length(Z0), fZ0(ii) = f(Z0(ii)); end
Z = [Z0 conj(Z0)]; fZ = [fZ0 conj(fZ0)];
FS = 'fontsize'; LW = 'linewidth'; MS = 'markersize';
IN = 'interpreter'; LT = 'latex'; PO = 'position';
mm = [3 7 13];
subplot(PO,[.07 .68 .4 .25])
loglog(imag(Z0),abs(fZ0),'k',LW,1), grid on
set(gca,FS,10,LW,1), grid on
title('function $f$ (degree 348)',FS,13,IN,LT), drawnow
xlabel('Im($z$)',FS,12,IN,LT), ylabel('$|f(z)|$',FS,12,IN,LT)
ylim([.04 40000]), set(gca,'ytick',10.^(0:2:4))
for im = 1:3
  if im==1, subplot(PO,[.57 .68 .4 .25]), end
  if im==2, subplot(PO,[.07 .29 .4 .25]), end
  if im==3, subplot(PO,[.57 .29 .4 .25]), end
  m = mm(im);
  [r,pol,res,zer,zj,fj,wj,errvec] = aaa(fZ,Z,'mmax',m);
  max_real_part_of_pole = max(real(pol))
  loglog(imag(Z0),abs(r(Z0)),LW,1), hold on
  set(gca,FS,10,LW,1), grid on
  xlabel('Im($z$)',FS,12,IN,LT), ylabel('$|r(z)|$',FS,12,IN,LT)
  s = int2str(m-1);
  title(['degree $' , s , '$ approximation'],FS,13,IN,LT), drawnow
  ylim([.04 40000]), set(gca,'ytick',10.^(0:2:4))
end
```

```
%% zetafun.m - AAA for analytic continuation of Riemann zeta function
zeta = @(z) sum((1e5:-1:1).^(-z),2);
Z = linspace(4-40i,4+40i).';
[r,pol,res,zeros] = aaa(zeta,linspace(4-40i,4+40i).');
clf, phaseplot(r,[-50 50 -50 50])
hold on, plot(Z,'.k','markersize',8), hold off
zeros

%% AAAspline - the AAA poles reveal the nodes of the spline
nodes = 0:10;
data = sin(nodes + nodes.^2/4);
s = chebfun.spline(nodes,data);
LW = 'linewidth'; MS = 'markersize'; FS = 'fontsize';
clf, subplot(211), plot(s,LW,1), grid on
hold on, plot(nodes,data,'.k',MS,12), ylim([-1.2 1.2]), hold off
title('Spline defined by data at 11 nodes',FS,15)
X = linspace(0,10,1000)';
[r,poles] = aaa(s(X),X,'mmax',200,'tol',1e-10);
subplot(212), plot([0 10],[0 0],'k',LW,.5), hold on
plot(poles,'.r',MS,8), hold off
grid on, axis equal, axis([0 10 -1.8 1.8])
title('AAA poles in the complex plane reveal singularities',FS,15)

%% equidata.m - AAA to fit equispaced data
ff = @(x) exp(x).*cos(10*x).*tanh(4*x);
grid = linspace(-1,1,40)'; data = ff(grid);
f = chebfun(ff); clf
LW = 'linewidth'; FS = 'fontsize'; MS = 'markersize';
subplot(311)
   plot(f,LW,1), hold on, plot(grid,data,'.k',MS,8), hold off, grid on
   title('f(x) sampled at 40 equispaced points',FS,11)
subplot(312)
   p = chebfun.interp1(grid,data);
   plot(p,LW,1), hold on, plot(grid,data,'.k',MS,8), hold off, grid on
   title('polynomial interpolant (degree 39)',FS,11)
subplot(313)
   [r,poles] = aaa(data,grid); pAAA = chebfun(r);
   rational_degree = length(poles)
   polynomial_degree = length(pAAA)-1
   plot(pAAA,LW,1), hold on, plot(grid,data,'.k',MS,8), hold off, grid on
   title('polynomial from AAA rational approx to the same data (degree 104)',FS,11)
   maxerror = norm(f-pAAA,inf)

%% conformaldemo.m - call conformal, which uses two AAA approximations
C = chebfun('exp(pi*1i*t)*(1+.4*cos(4*pi*t))','trig');
C = real(C) + .7i*imag(C);
clf, [f,finv] = conformal(C,'plots');
Z = 2*(rand(2e6,1)+1i*rand(2e6,1)) - (1+1i);
Z = Z(abs(Z)<1); Z = Z(1:1e6);
tic, W = finv(Z); Z2 = f(W); t = toc;
disp('Time to map one million points back and forth:'), disp(t)
disp('Maximum error in that process:'), disp(norm(Z-Z2,inf))

%% AAAeigs.m - find eigenvalue via AAA fit to scalarized resolvent
rng(1); [U,~] = qr(randn(80)+1i*randn(80));
e = (-19.5:19.5); e = [e 1i*e]; A = U*diag(e)/U;
u = randn(80,1); v = randn(80,1);
npts = 100; Z = exp(2i*pi*(1:npts)'/npts); F = 0*Z; I = eye(80);
for j = 1:npts, F(j) = u'*((Z(j)*I-A)\v); end
[r,pol] = aaa(F,Z);
pol(abs(pol)<1)
LW = 'linewidth'; FS = 'fontsize'; MS = 'markersize';
clf, plot([0 0],[-5 5],'k',LW,.3), hold on, plot([-5 5],[0 0],'k',LW,.3)
plot(pol,'.r',MS,12), grid on, axis(5*[-1 1 -1 1]), axis square
plot(Z,'.k',MS,5), hold off
set(gca,'xtick',-4.5:4.5,'ytick',-4.5:4.5,FS,7)
title('Matrix eigenvalues via AAA fit to scalarized resolvent',FS,15)
```