Paper: VEXPA: Validated EXPonential Analysis through regular sub-sampling.pdf

Matteo Briani, Annie Cuyt and Wen-shin Lee

VEXPA: Validated EXPonential Analysis through regular sub-sampling

Example Figure 5.

Contents

Script environment

This script does not depend on the random number generator state.

```
clear
close all
```

Example Figure 5.

A problem that may occur when decimation causes aliasing is the possible collision of frequencies. For instance, two distinct eigenvalues λ_1 and λ_2 may be aliased to the same eigenvalue ${}_u\lambda_1 = {}_u\lambda_2$. However unlikely, we want to discuss how to deal with this situation. We explain the remedy on an example.

Let $\phi(t)$ be specified by $n = 2, \alpha_1 = \alpha_2 = 1, \mu_1 = 2\pi\mathrm{i}13, \mu_2 = 2\pi\mathrm{i}33$. We set $\Omega = 100$ and consider one sample $\phi_j = \phi(j/\Omega)$ every ten samples ($u = 10$) thus changing Ω to be 10. Due to aliasing, λ_1 and λ_2 are mapped to another location in the complex plane. In particular, we have

${}_u\lambda_1 = {}_u\lambda_2 = \exp(2\pi\mathrm{i}3/10)$,

because

$\exp(2\pi\mathrm{i}33/10) = \exp(2\pi\mathrm{i}13/10) = \exp(2\pi\mathrm{i}3/10)$.

```
n = 2;

alpha = [1,1];
mu = 2*pi*1i*[13,33];
r = 1/100;
```

```
u = 10;

fprintf('Aliased eigenvalues:')
ulambda = exp(mu*r*10)
```

```
Aliased eigenvalues:
ulambda =

  Column 1

 -0.309016994374947 + 0.951056516295154i

  Column 2

 -0.309016994374950 + 0.951056516295153i
```

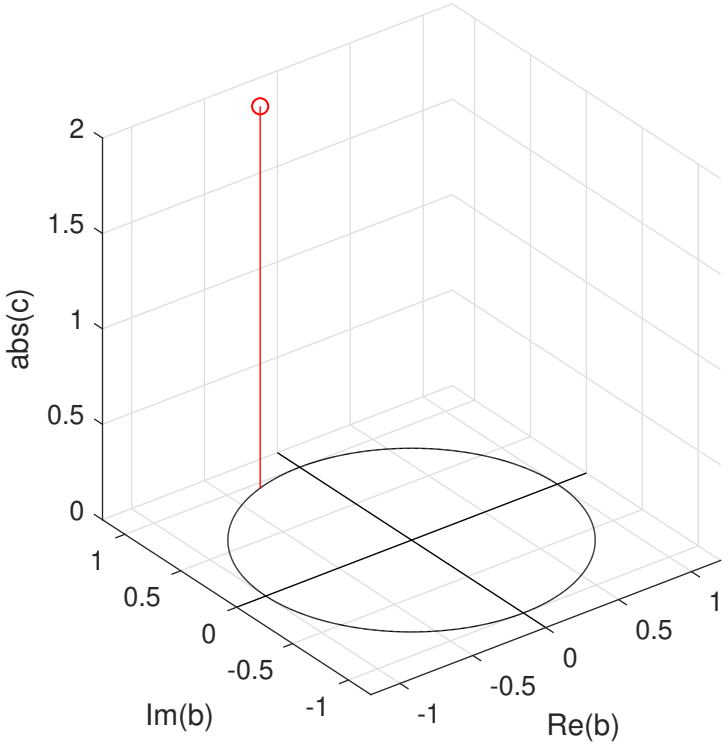
So in the decimation step (9) Prony s method retrieves a single frequency with associated coefficient $\alpha_1 + \alpha_2$.

```
params = MultiExponentialParameters(1/r,{exp(mu*r),alpha},...
                                    'normalized');
signal = params.construct(1000);

bcsolver = MultiExponentialSolver...
            (BSolverEsprit('--nsamples',2*n,'--ncols',n,...
                           '--nrows',n+1,'--nterms',n),...
            CSolverVandermondeLS('--nrows',2*n,'--delta',1),...
            '--nsamples',2*n);

params_u = bcsolver.solve(signal.select(1:u:1+(2*n-1)*u));
params_u.remove(abs(params_u.c) < 1e-3);

plot3_base_terms(params_u);
title(['Retrieved single frequency with coefficient '...
       '\alpha_1 + \alpha_2.'])
```



It is however still possible to retrieve the original values λ_1 and λ_2 in the recovery step. As explained, the generalized eigenvalue ${}_u\lambda_1 = {}_u\lambda_2$ stands for a set of values $U_1 = U_2$ that now contains both the correct λ_1 and λ_2. We choose s coprime with u and compute the values ${}^{ms}\alpha_1$ (remember that the computed ${}^{0}\alpha_1 = 2$ now equals the sum of the true coefficients).

```
M = 8;
s = 3;

csolver = CSolverVandermondeLS('--nrows',n,'--delta',1);

alpha_ms = zeros(numel(params_u.c),M);
alpha_ms(:,1) = params_u.c;

for k = 1:7
    alpha_ms(:,k+1) = csolver.solve(...
                    signal.select(1+k*s:u:1+k*s+2*u),params_u.b);
end
```

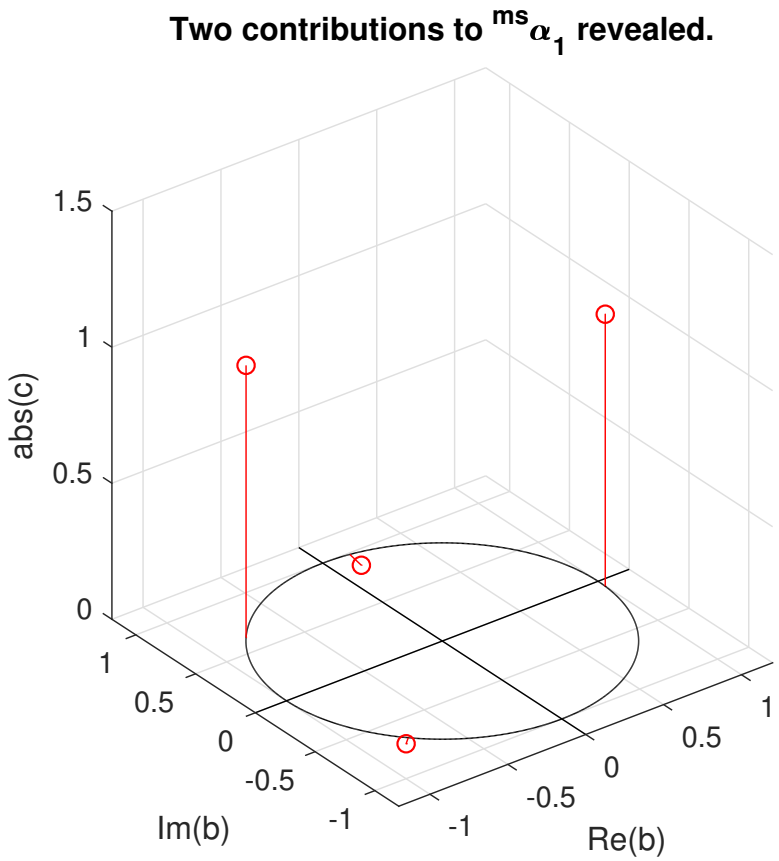
Since s is coprime with u, no frequency collision occurs in ${}^{ms}\alpha_1$ which is following the model

${}^{ms}\alpha_1 = \alpha_1 \exp(\mu_1 ms\Delta) + \alpha_2 \exp(\mu_2 ms\Delta)$, $m = 0, \ldots, M-1$.

So in the analysis of (13) Prony s method reveals two contributions ${}^{s}\lambda_1$ and ${}^{s}\lambda_2$.

```
bcsolver = MultiExponentialSolver...
            (BSolverEsprit('--nsamples',M,'--ncols',M/2,...
                           '--nrows',M/2+1,'--nterms',M/2),...
            CSolverVandermondeLS('--nrows',M,'--delta',1),...
            '--nsamples',M);

params_ms = bcsolver.solve(Signal(1,alpha_ms));
plot3_base_terms(params_ms);
title('Two contributions to ^{ms}\alpha_1 revealed.')
```



These bring forth the sets S_1 and S_2 respectively containing λ_1 and λ_2. The intersections $U_1 \cap S_1$ and $U_2 \cap S_2 = U_1 \cap S_2$ reveal the original λ_1 and λ_2.

```
U1 = power(params_u.b, 1/u) * exp(2*pi*1i*(0:u-1)/u);
```

```
S1 = power(params_ms.b(1), 1/s) * exp(2*pi*1i*(0:s-1)/s);
S2 = power(params_ms.b(2), 1/s) * exp(2*pi*1i*(0:s-1)/s);

figure
plot_unit_circle;
hold on
plot(U1,'bo','MarkerSize',10,'LineWidth',2);
plot(S1,'gs','MarkerSize',10,'LineWidth',2);
plot(S2,'g^','MarkerSize',10,'LineWidth',2);
plot(exp(mu*r),'rs','MarkerSize',20,'LineWidth',3);
title(['Figure 5. (left) Identifying U_1\cap S_1={\lambda_1} '...
       'and U_2\cap S_2=U_1\cap S_2={\lambda_2} from u=10 '...
       '(red squares).'])

figure
bar(1:4,abs(params_ms.c),0.05,'g');
hold on
plot(1,0.03+abs(params_ms.c(1)),'gs','MarkerSize',20);
plot(2,0.03+abs(params_ms.c(2)),'g^','MarkerSize',20);
title(['Figure 5. (right) The |\alpha_1| of ^s\lambda_1 '...
       '(green triangle) and |\alpha_2| of ^s\lambda_2 '...
       '(green square) for s = 3, M = 8.'])
```

Figure 5. (left) Identifying $U_1 \cap S_1 = \lambda_1$ and $U_2 \cap S_2 = U_1 \cap S_2 = \lambda_2$ from u=10 (red squares).

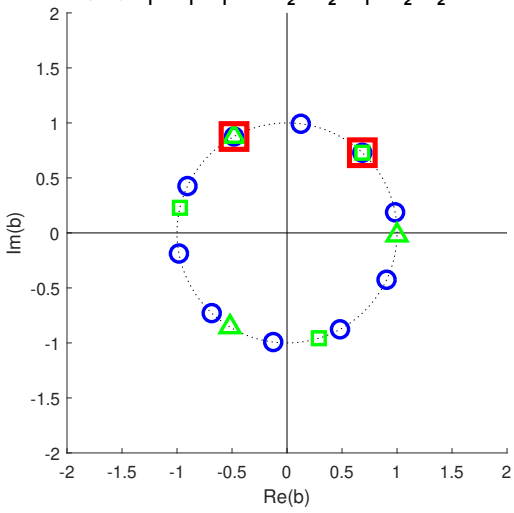


Figure 5. (right) The $|\alpha_1|$ of ${}^s\lambda_1$ (green triangle) and $|\alpha_2|$ of ${}^s\lambda_2$ (green square) for s = 3, M = 8.