Largest Eigenvalue Distributions

Per-Olof Persson

Tue Nov 26, 2002

Topics

- Distribution of largest eigenvalue of random Gaussian matrices
- Numerical solution of the Painlevé II nonlinear differential equation
- General β
- Perturbation theory

Random Matrix Eigenvalues

- Hermitian $N \times N$ matrix A, diagonal elements x_{jj}, upper triangular elements $x_{jk} = u_{jk} + iv_{jk}$, independent, zero-mean Gaussians, and

$$\begin{cases} \mathrm{Var}(x_{jj}) = 1, & 1 \le j \le N \\ \mathrm{Var}(u_{jk}) = \mathrm{Var}(v_{jk}) = \frac{1}{2}, & 1 \le j < k \le N \end{cases}$$

- MATLAB:

```
A=randn(N)+i*randn(N);
A=(A+A')/2;
```

- Scaling of largest eigenvalue:

$$\lambda'_{\max} = n^{\frac{1}{6}} \left(\lambda_{\max} - 2\sqrt{n}\right)$$

"Brute-force" solution

- ```
  for ii=1:trials
    A=randn(N)+i*randn(N);
    A=(A+A')/2;
    lmax=max(eig(A));
    lmaxscaled=n^(1/6)*(lmax-2*sqrt(n));
    % Store lmax
  end
  ```
- Requires n^2 memory $\Longrightarrow n < 10^4$
- Requires n^3 computations per sample $\Longrightarrow$ Many days to get a nice histogram
  ```

Faster Method

- Real, symmetric, tridiagonal matrix with the same eigenvalues as the previous matrix for $\beta = 2$ (Dumitriu, Edelman):

$$H_\beta \sim \frac{1}{\sqrt{2}} \begin{pmatrix} N(0,2) & \chi_{(n-1)\beta} & & & \\ \chi_{(n-1)\beta} & N(0,2) & \chi_{(n-2)\beta} & & \\ & \ddots & \ddots & \ddots & \\ & & \chi_{2\beta} & N(0,2) & \chi_\beta \\ & & & \chi_\beta & N(0,2) \end{pmatrix}$$

- Requires n memory, largest eigenvalue can be computed in n time using Arnoldi iterations or method of bisection.
- MATLAB: Built-in function `eigs` or interface to LAPACK function `DSTEBZ` implemented in mex-file `maxeig`

Even Faster Method

- When computing the largest eigenvalue, the matrix is well approximated by the upper-left n_{cutoff} by n_{cutoff} matrix, where

$$n_{\text{cutoff}} \approx 10n^{\frac{1}{3}}$$

- $\Longrightarrow$ very large n can be used ($> 10^{12}$)
- Also, $\chi_n^2 \approx n +$ Gaussian for large n

leig.m

```
ls=leigsub(1e6,1e4,2);
histdistr(ls,-7:0.2:3);
```

leigsub.m

```
function ls=leigsub(n,nrep,beta)

cutoff=round(10*n^(1/3));
d1=sqrt(n-1:-1:n+1-cutoff)'/2/sqrt(n);

ls=zeros(1,nrep);
for ii=1:nrep
  d0=randn(cutoff,1)/sqrt(n*beta);
  l=maxeig(d0,d1);

  ls(ii)=l;
end

ls=(ls-1)*n^(2/3)*2;
```

Largest Eigenvalue Distribution

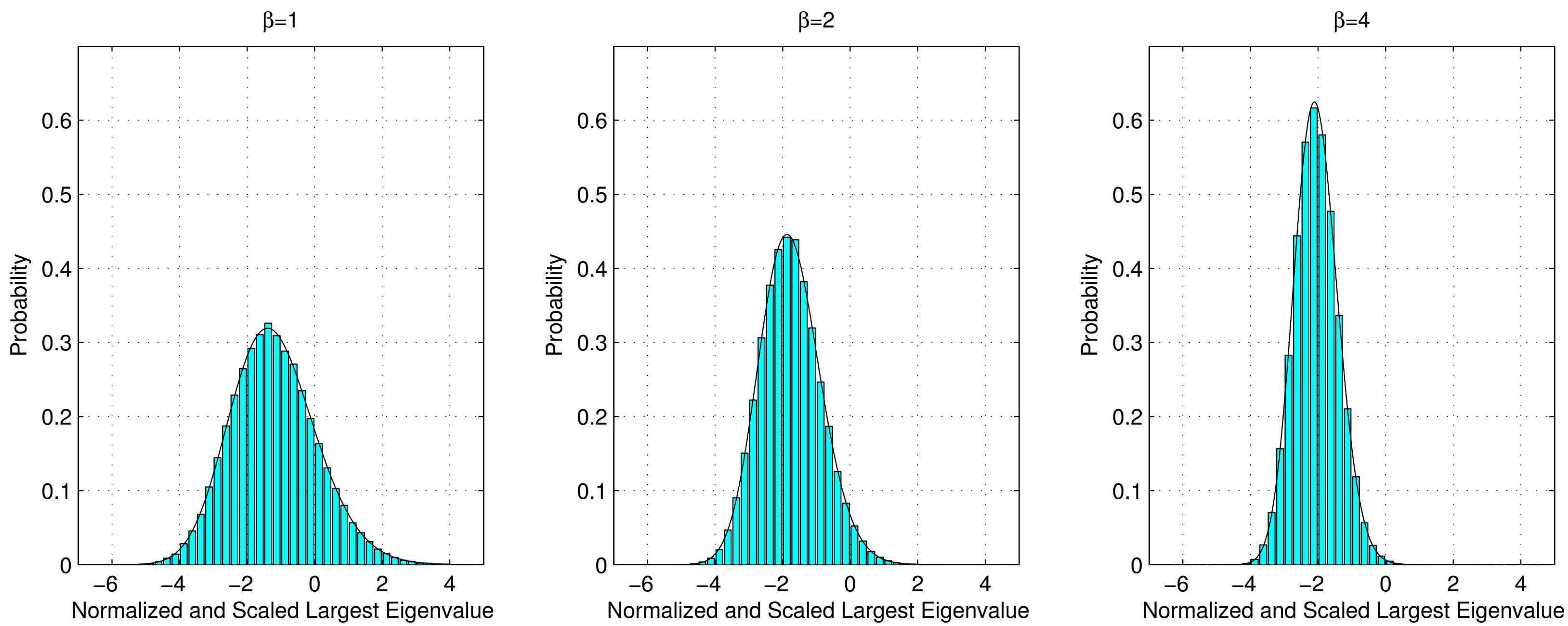


Figure 1: Probability distribution of scaled largest eigenvalue (10^5 repetitions, $n = 10^9$)

Differential Equation for Distributions

- (Tracy, Widom) Probability distribution $f_2(s)$ is given by

$$f_2(s) = \frac{d}{ds} F_2(s)$$

where

$$F_2(s) = \exp\left(- \int_s^{\infty} (x - s) q(x)^2 \, dx \right)$$

and $q(s)$ satisfies the Painlevé II differential equation:

$$q'' = sq + 2q^3$$

with the boundary condition

$$q(s) \sim \mathrm{Ai}(s), \qquad \text{as } s \to \infty$$

Distributions for $\beta = 1$ and $\beta = 4$

- The distributions for $\beta = 1$ and $\beta = 4$ can be computed from $F_2(s)$ as

$$F_1(s)^2 = F_2(s) e^{-\int_s^\infty q(x)\,dx}$$

$$F_4\left(\frac{s}{2^{\frac{2}{3}}}\right)^2 = F_2(s)\left(\frac{e^{\frac{1}{2}\int_s^\infty q(x)\,dx} + e^{-\frac{1}{2}\int_s^\infty q(x)\,dx}}{2}\right)^2$$

Numerical Solution

- Write as 1^{st} order system:

$$\frac{d}{ds}\begin{pmatrix} q \\ q' \end{pmatrix} = \begin{pmatrix} q' \\ sq + 2q^3 \end{pmatrix}$$

- Solve as initial-value problem starting at $s = s_0 =$ large positive number

- Initial values (boundary conditions):

$$\begin{cases} q(s_0) &= \mathrm{Ai}(s_0) \\ q'(s_0) &= \mathrm{Ai}'(s_0) \end{cases}$$

- Explicit ODE solver (RK4)

Post-processing

- $F_2(s) = \exp\left(-\int_s^\infty (x-s)q(x)^2\,dx\right)$ could be computed using high-order quadrature

- Convenient trick: Set $I(s) = \int_s^\infty (x-s)q(x)^2\,dx$ and differentiate:

$$I'(s) = -\int_s^\infty q(x)^2\,dx$$
$$I''(s) = q(s)^2$$

  Add these equations and the variables $I(s), I'(s)$ to ODE system, and let the solver do the integration

- Also, add variable $J(s) = \int_s^\infty q(x)\,dx$ and equation $J'(s) = q(s)$ for computation of $F_1(s)$ and $F_4(s)$

- $f_\beta(s) = \frac{d}{ds}F_\beta(s)$ using numerical differentiation

mkfbeta.m

```
deq=inline('[y(2); s*y(1)+2*y(1)^3; y(4); y(1)^2; y(1)]','s','y');
opts=odeset('reltol',1e-13,'abstol',1e-14);

t0=5; tn=-8;
tspan=linspace(t0,tn,1000);
y0=[airy(t0);airy(1,t0);0;airy(t0)^2;0];

[s,y]=ode45(deq,tspan,y0,opts);

F2=exp(-y(:,3));
F1=sqrt(F2.*exp(y(:,5)));
F4=sqrt(F2).*(exp(-y(:,5)/2)+exp(y(:,5)/2))/2;
s4=s/2^(2/3);

f2=gradient(F2,s);
f1=gradient(F1,s);
f4=gradient(F4,s4);

plot(s,f1,s,f2,s4,f4)
legend('\beta=1','\beta=2','\beta=4')
xlabel('s')
ylabel('f_\beta(s)','rotation',0)
grid on
```

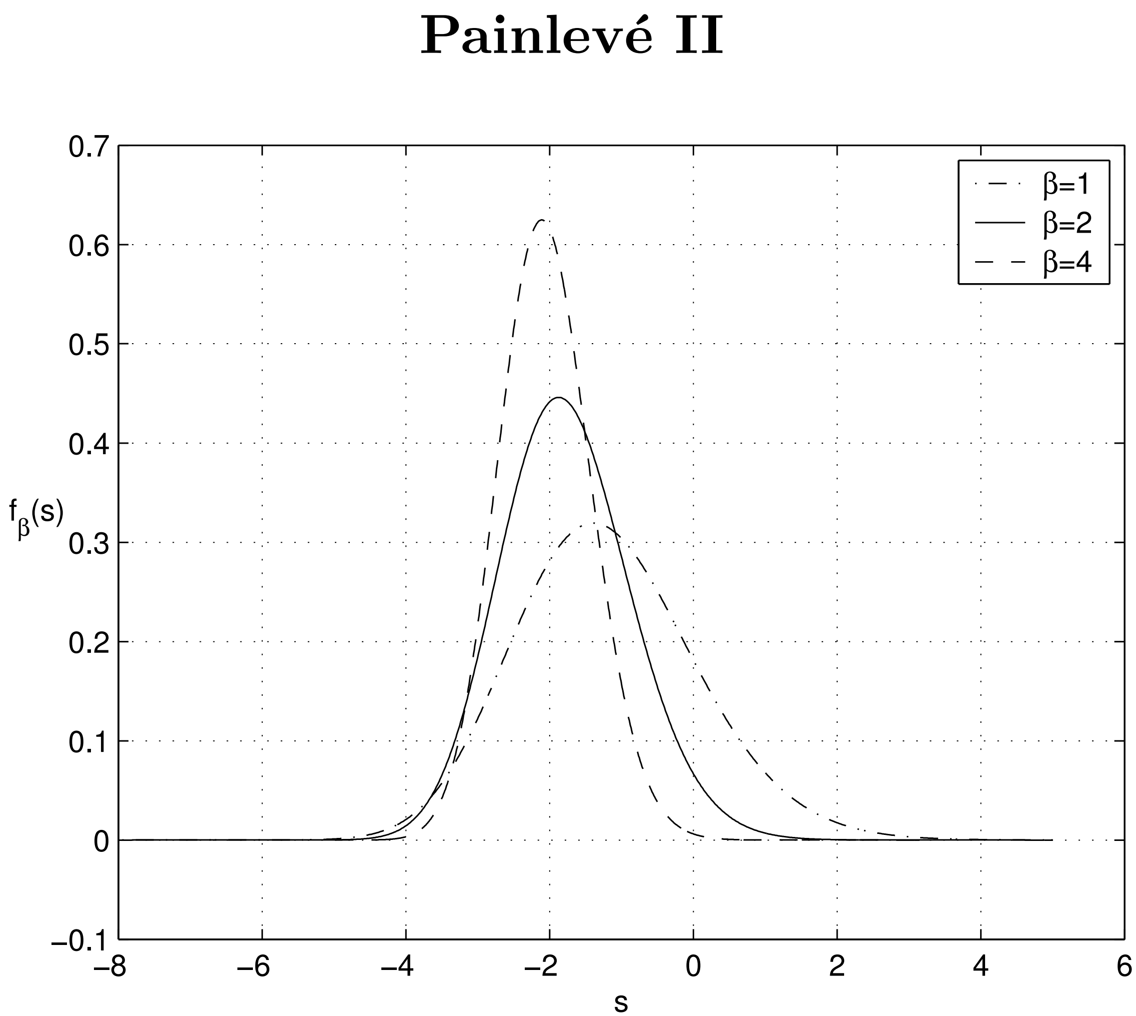


Figure 2: The probability distributions $f_1(s)$, $f_2(s)$, and $f_4(s)$, computed from Painlevé II solution.

General β

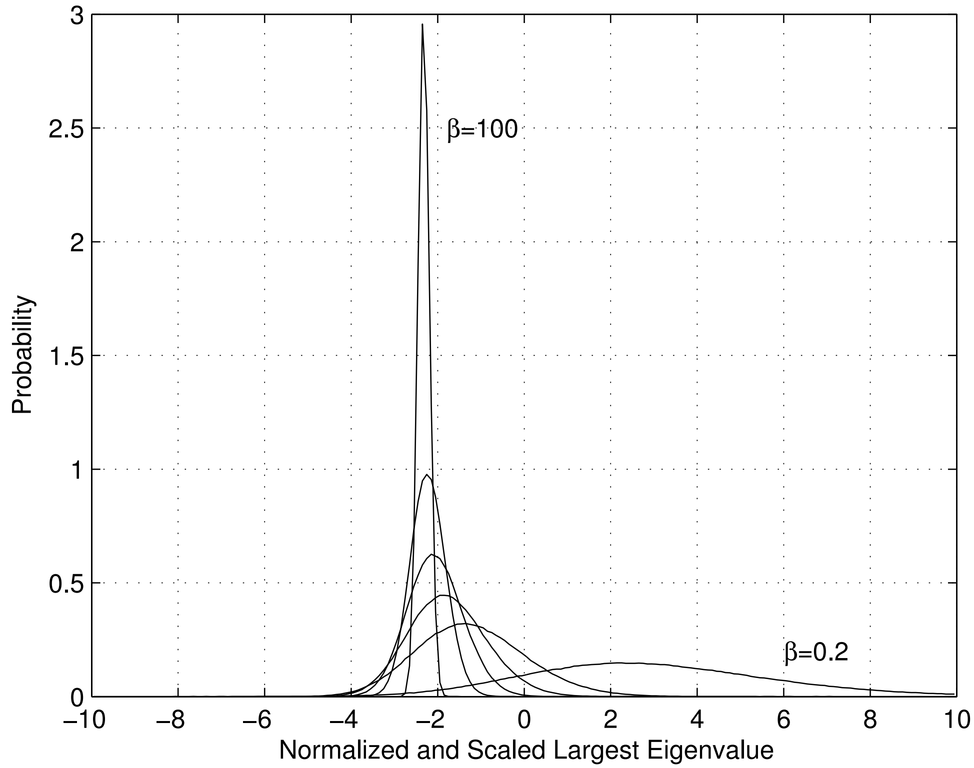


Figure 3: Probability distribution of scaled largest eigenvalue for $\beta = 100, 10, 4, 2, 1, 0.2$.

General β

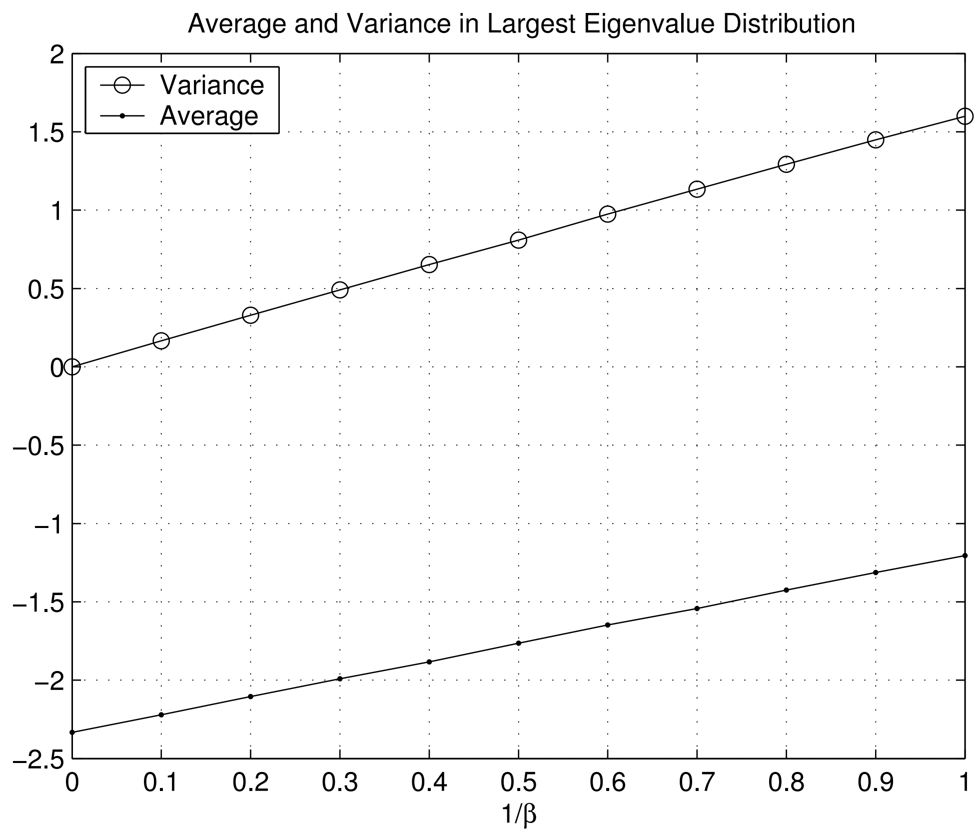


Figure 4: Average and variance of probability distribution as a function of $1/\beta$.

Convection-Diffusion Approximation

- Average and variance linear in $t = 1/\beta$ suggest constant-coefficient convection-diffusion approximation:

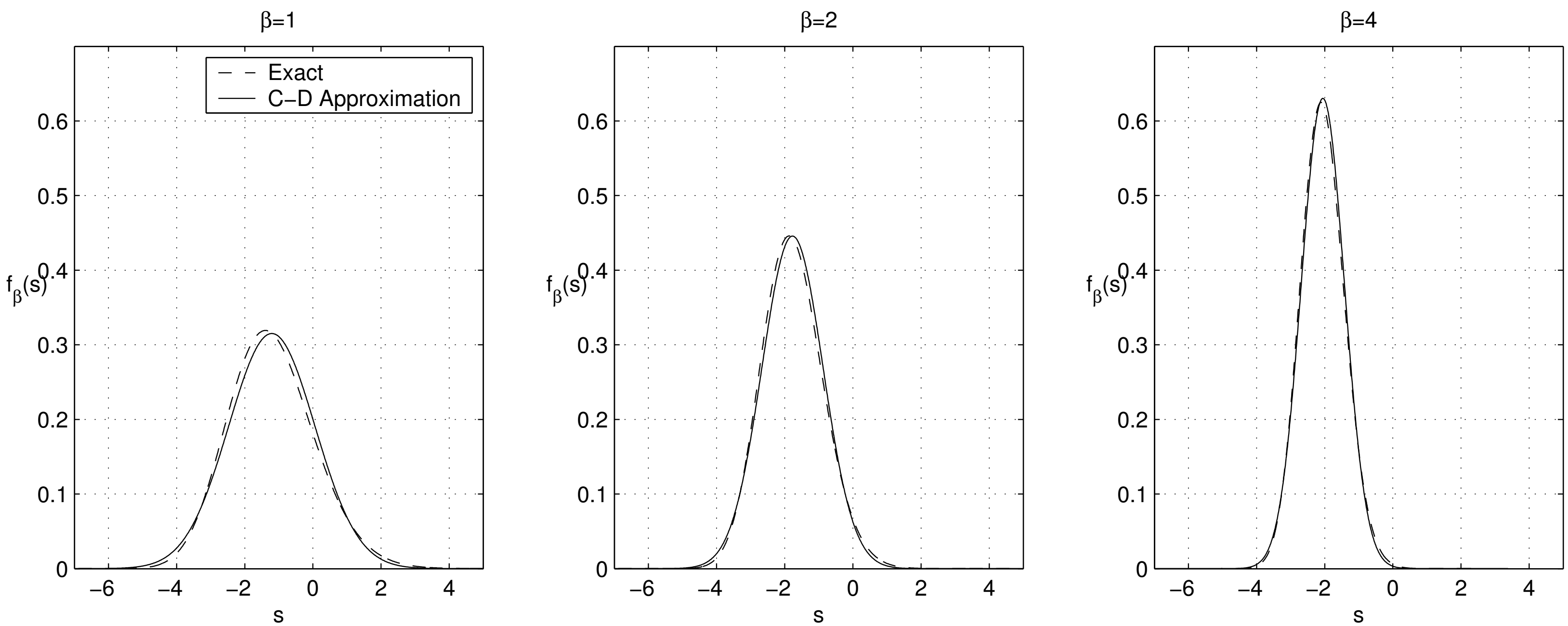
$$\frac{df}{dt} = \frac{C_2}{2}\frac{d^2 f}{ds^2} - C_1 \frac{df}{ds}$$
$$f(0,s) = \delta(s - \mathrm{AiZ}_1)$$

where C_1, C_2 are the slopes of the average and the variance.

- Solution:

$$f(t,s) = \frac{1}{\sqrt{2\pi C_2 t}} \exp\left(-\frac{(x - \mathrm{AiZ}_1 - C_1 t)^2}{2C_2 t}\right)$$

Convection-Diffusion Approximation



Perturbation Theory

- Split the tridiagonal matrix:

$$T_\beta^n = T_\infty^n + \frac{2}{\sqrt{\beta}} \cdot (\text{diagonal Gaussian})$$

$$= \frac{1}{2\sqrt{n}} \begin{bmatrix} 0 & \sqrt{n-1} & & \\ \sqrt{n-1} & 0 & \ddots & \\ & \ddots & 0 & \sqrt{1} \\ & & \sqrt{1} & 0 \end{bmatrix} + \frac{2}{\sqrt{\beta}} \begin{bmatrix} G & & & \\ & G & & \\ & & \ddots & \\ & & & G \end{bmatrix}$$

- Define:

$$T_\beta^{n,k} = Q_{n,k}^T T_\beta^n Q_{n,k}$$

where

$$Q_{n,k} = \begin{bmatrix} q_{n,1} & q_{n,2} & \dots & q_{n,k} \end{bmatrix} = k \text{ first eigenvectors of } T_\infty^{n,k}$$

Perturbation Theory

- T_∞^n is diagonalized, the diagonal Gaussian matrix becomes a full matrix:

$$T_\beta^{n,k} = Q_{n,k}^T T_\beta^n Q_{n,k} = \begin{bmatrix} \eta_1 & & & \\ & \eta_2 & & \\ & & \ddots & \\ & & & \eta_k \end{bmatrix} + \frac{2}{\sqrt{\beta}} \begin{bmatrix} G & G & \dots & G \\ G & G & \dots & G \\ \vdots & \vdots & \ddots & \vdots \\ G & G & \dots & G \end{bmatrix}$$

where the covariance matrix for the Gaussians is

$$C_{ij,i'j'} = \sum q_{n,i} q_{n,j} q_{n,i'} q_{n,j'}$$

Perturbation Theory

- When $n \to \infty$,

$$\eta_i = \mathrm{AiZ}_i$$

$$q_{\infty,i}(x) = \frac{\mathrm{Ai}(x + \mathrm{AiZ}_i)}{\sqrt{\int_0^\infty \mathrm{Ai}(x + \mathrm{AiZ}_i)^2}} = \frac{\mathrm{Ai}(x + \mathrm{AiZ}_i)}{\mathrm{Ai}'(\mathrm{AiZ}_i)}$$

$$C_{ij,i'j'} = \int_0^\infty q_{\infty,i}(x) q_{\infty,j}(x) q_{\infty,i'}(x) q_{\infty,j'}(x)\, dx$$

mkC.m

```
function C=mkC(k)

C=zeros(k^2,k^2);
for ii=1:k
  for jj=1:k
    for kk=1:k
      for ll=1:k
        f=['airy(x+AiZ(',int2str(ii),')).*' ...
           'airy(x+AiZ(',int2str(jj),')).*' ...
           'airy(x+AiZ(',int2str(kk),')).*' ...
           'airy(x+AiZ(',int2str(ll),'))'];
        g=['airy(1,AiZ(',int2str(ii),')).*' ...
           'airy(1,AiZ(',int2str(jj),')).*' ...
           'airy(1,AiZ(',int2str(kk),')).*' ...
           'airy(1,AiZ(',int2str(ll),'))'];
        func=inline(['real((',f,')./(',g,'))']);
        c=quadl(func,0,10,1e-4);
        C(ii+(jj-1)*k,kk+(ll-1)*k)=c;
      end
    end
  end
end
```

mkCnum.m

```
function C=mkCnum(k)

n=100;

d1=sqrt(n-1:-1:1)'/2/sqrt(n);
A=diag(d1,1)+diag(d1,-1);
[V,D]=eig(A);

q=[V(:,end:-1:end-k+1)];

C=zeros(k^2,k^2);
for ii=1:k
  for jj=1:k
    for kk=1:k
      for ll=1:k
        C(ii+(jj-1)*k,kk+(ll-1)*k)=sum(q(:,ii).*q(:,jj).*q(:,kk).*q(:,ll));
      end
    end
  end
end
C=C*n^(1/3);
```

covsub.m

```
function ls=covsub(nrep,beta,R)

k=sqrt(size(R,1));
A=diag(AiZ(1:k));

ls=zeros(1,nrep);
for ii=1:nrep
  l=max(eig(A+1/sqrt(beta)*2*reshape(R*randn(k^2,1),k,k)));
  ls(ii)=l;
end
```

covcmp.m

```
n=1e6;
nrep=1e4;
k=2;

invbs=0:0.1:1;
bs=1./invbs;
nbs=length(bs);

ms1=zeros(1,nbs);
stds1=zeros(1,nbs);

C=mkC(k);
R=chol(C)';
for ib=1:nbs
  beta=bs(ib);

  ls=covsub(nrep,beta,R);
  ms1(ib)=mean(ls);
  stds1(ib)=std(ls);
end

plot(invbs,AiZ(1)+1.1315*invbs,invbs,ms1, ...
     invbs,1.6025*invbs,invbs,stds1.^2)
```

Perturbation Theory