Numerical Evaluation of Standard Distributions in Random Matrix Theory

A Review of Folkmar Bornemann's MATLAB Package and Paper

Matt Redmond

Department of Mathematics
Massachusetts Institute of Technology
Cambridge, MA 02139, USA

May 15, 2013

Level Spacing Function

Level Spacing Function

Definition (Gaussian Ensemble Spacing Function)

Let $J \subset \mathbb{R}$ be an open interval.

$$E_\beta^{(n)}(k; J) \equiv \mathbb{P}(k \text{ eigenvalues of the } n \times n \text{ Gaussian } \beta\text{-ensemble lie in } J)$$

Level Spacing Function

Definition (Gaussian Ensemble Spacing Function)

Let $J \subset \mathbb{R}$ be an open interval.

$$E_\beta^{(n)}(k; J) \equiv \mathbb{P}(k \text{ eigenvalues of the } n \times n \text{ Gaussian } \beta\text{-ensemble lie in } J)$$

Let $J = (0, s)$. Then $E_2(0; J) =$ probability no eigenvalues lie in $(0, s)$.

Fredholm Determinant

Fredholm Determinant

It is well known that $E_2(0; J)$ can be represented as a Fredholm determinant:

Fredholm Determinant

It is well known that $E_2(0; J)$ can be represented as a Fredholm determinant:

Theorem (Gaudin 1961)

Given $K_{\sin}(x, y) = sinc(\pi(x - y))$,

$$E_2(0, J) = \det\left(I - K_{\sin} \restriction_{L^2_J}\right)$$

Note the operator's restriction to square integrable functions over J. In general we will choose $J = (0, s)$, and will notate $E_2(0, (0, s))$ as $E_2(0, s)$ as per Bornemann's conventions.

Integral Formulation

Theorem (Jimbo, Miwa, Mori, Sato 1980)

$$E_2(0; s) = \exp\left(-\int_0^{\pi s} \frac{\sigma(x)}{x}\, dx\right)$$

where $\sigma(x)$ solves a particular form of the Painleve V equation:

$$(x\sigma'')^2 = 4(\sigma - x\sigma')(x\sigma - \sigma - (\sigma')^2), \quad \sigma(x) \approx \frac{x}{\pi} + \frac{x^2}{\pi^2} \quad (x \to 0)$$

Tracy-Widom Distribution

Tracy-Widom Distribution

Definition (Tracy-Widom Distribution)

Let $F_2(s) \equiv \mathbb{P}($ no eigenvalues of large-matrix limit GUE lie in $(s, \infty))$

Determinantal Representation

Determinantal Representation

Theorem (Bronk 1964)

Given

$$K_{Ai}(x,y) = \frac{Ai(x)Ai'(y) - Ai'(x)Ai(y)}{x-y}$$

we have

$$F_2(s) = \det\left(I - K_{Ai} \restriction_{L^2_{(s,\infty)}}\right)$$

Integral Formulation

Integral Formulation

Theorem (Tracy, Widom 1993)

$$F_2(s) = \exp\left(-\int_s^\infty (x-s)u(x)^2 dx\right)$$

where $u(x)$ is the Hastings-McLeod (1980) solution to the Painleve II equation

$$u'' = 2u^3 + xu, \quad u(x) \approx Ai(x) \quad (x \to \infty)$$

The common point of view, and why it's wrong.

The common point of view, and why it's wrong.

Common point of view:

- Painleve formulation is somehow numerically "better behaved" than Fredholm determinant
- Solving initial value problem for numerical integration is easier to implement than Fredholm

The common point of view, and why it's wrong.

Common point of view:

- Painleve formulation is somehow numerically "better behaved" than Fredholm determinant
- Solving initial value problem for numerical integration is easier to implement than Fredholm

Bornemann's view:

- Numerical evaluation of Painleve transcendents is actually fairly involved. Stability is a major concern.
- There exists a simple, fast, accurate numerical method for evaluating Fredholm determinants
- Many multivariate functions (joint prob. dists.) have a nice representation as a Fredholm determinant, but no representation in terms of a nonlinear PDE.

Straightforward Approach: Solving the IVP for Painleve

Straightforward Approach: Solving the IVP for Painleve

All of the examples we are interested in take *asymptotic IVP form*:

Straightforward Approach: Solving the IVP for Painleve

All of the examples we are interested in take *asymptotic IVP form*:
Given an interval (a, b), we seek $u(x)$ that solves

$$u''(x) = f(x, u(x), u'(x))$$

subject to either of the asymptotic one-sided conditions

$$u(x) \approx u_a(x) \quad (x \to a)$$

or

$$u(x) \approx u_b(x) \quad (x \to b)$$

Straightforward Approach (cont.)

Straightforward Approach (cont.)

Problem 1: Must identify asympotic expansion of $u(x)$ – not always easy.

Straightforward Approach (cont.)

Problem 1: Must identify asympotic expansion of $u(x)$ – not always easy. Even with expansion, we must choose initial point and asymptotic order of approximation, so choose $a_+ > a$ (or $b_- < b$) close to boundary and compute solution to the (standard) IVP problem

$$v''(x) = f(x, v(x), v'(x))$$

$$v(a_+) = u_a(a_+), \quad v'(a_+) = u_a'(a_+)$$

or

$$v(b_-) = u_b(b_-), \quad v'(b_-) = u_b'(b_-)$$

Straightforward Approach (cont.)

Straightforward Approach (cont.)

Problem 2: Standard solution methods demonstrate numerical instability.

Straightforward Approach (cont.)

Problem 2: Standard solution methods demonstrate numerical instability.
Example: Computing $F_2(s) = \exp\left(-\int_s^\infty (x-s)u(x)^2 dx\right)$:

$$v(x)'' = 2v(x)^3 + xv(x), \quad v(b_-) = \text{Ai}(b_-), \quad v'(b_-) = \text{Ai'}(b_-)$$

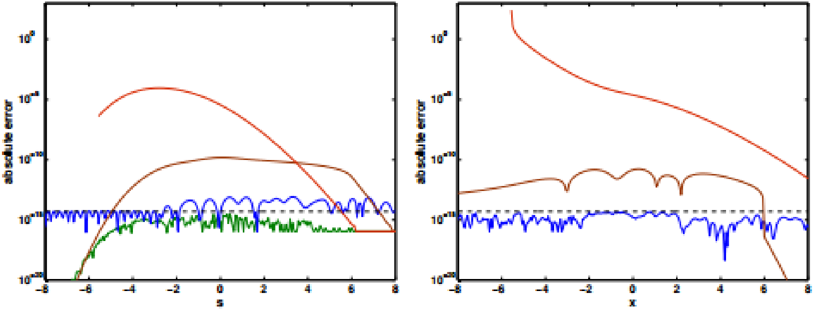
Straightforward Approach (cont.)

Problem 2: Standard solution methods demonstrate numerical instability.
Example: Computing $F_2(s) = \exp\left(-\int_s^\infty (x-s)u(x)^2 dx\right)$:

$$v(x)'' = 2v(x)^3 + xv(x), \quad v(b_-) = \text{Ai}(b_-), \quad v'(b_-) = \text{Ai'}(b_-)$$

Choosing $b_- \geq 8$ gives initial values accurate to machine precision (about 10^{-16} for IEEE doubles). Choose $b_- = 12$ yields these results:

Stability Issues



a. error in evaluating $F_2(s)$ b. error in evaluating $u(x)$

method	reference	max. error	run time
IVP/Matlab's ode45	Edelman and Persson (2005)	$9.0 \cdot 10^{-5}$	11 sec
BVP/Matlab's bvp4c	Dieng (2005)	$1.5 \cdot 10^{-10}$	3.7 sec
BVP/spectral colloc.	Driscoll et al. (2008)	$8.1 \cdot 10^{-14}$	1.3 sec
Fredholm determinant	Bornemann (2010a)	$2.0 \cdot 10^{-15}$	0.69 sec

Less Straightforward Approach: Solving the BVP for Painleve

Stability issues described in depth in Bornemann's paper lead to a BVP approach.
We use asymptotic expression $u_a(x)$ at $(x \to a)$ to infer asymptotic expression $u_b(x)$ at $(x \to b)$, or vice versa.
Approximate $u(x)$ by solving BVP:

$$v''(x) = f(x, v(x), v'(x)), \quad v(a_+) = u_a(a_+), \quad v(b_-) = u_b(b_-)$$

Requires four choices: values of a_+, b_-, and order of asymptotic accuracy for $u_a(x)$ and $u_b(x)$

Less Straightforward Approach: An Example

Less Straightforward Approach: An Example

Computing $F_2(s)$ via computation of $u(x)$ via BVP methods:

Less Straightforward Approach: An Example

Computing $F_2(s)$ via computation of $u(x)$ via BVP methods:
By definition, $u(x) \approx \text{Ai}(x) \quad (x \to \infty)$ so we take $u_b(x) = \text{Ai}(x)$.
Choose $a_+ = -10$, $b_- = 6$ (Dieng, 2005).

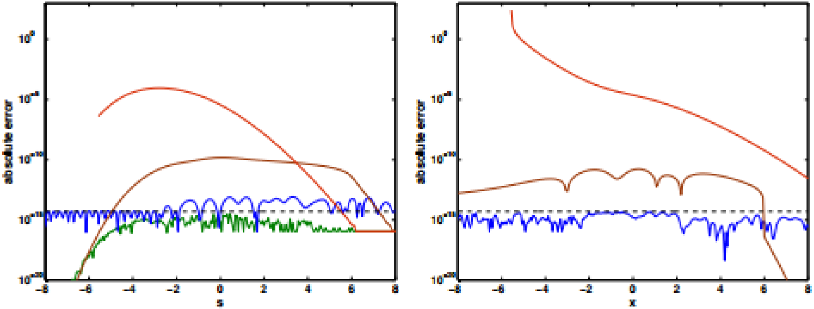
Less Straightforward Approach: An Example

Computing $F_2(s)$ via computation of $u(x)$ via BVP methods:
By definition, $u(x) \approx \mathrm{Ai}(x) \quad (x \to \infty)$ so we take $u_b(x) = \mathrm{Ai}(x)$.
Choose $a_+ = -10$, $b_- = 6$ (Dieng, 2005).
We need to choose a sufficiently accurate asymptotic expansion for $u_a(x)$. Tracy and Widom show

$$u(x) = \sqrt{-\frac{x}{2}} \left(1 + \frac{1}{8} x^{-3} - \frac{73}{128} x^{-6} + \frac{10657}{1024} x^{-9} + O(x^{-12}) \right), \quad (x \to -\infty)$$

so we'll use that for $u_a(x)$.

Stability Issues



a. error in evaluating $F_2(s)$ b. error in evaluating $u(x)$

method	reference	max. error	run time
IVP/Matlab's `ode45`	Edelman and Persson (2005)	$9.0 \cdot 10^{-5}$	11 sec
BVP/Matlab's `bvp4c`	Dieng (2005)	$1.5 \cdot 10^{-10}$	3.7 sec
BVP/spectral colloc.	Driscoll et al. (2008)	$8.1 \cdot 10^{-14}$	1.3 sec
Fredholm determinant	Bornemann (2010a)	$2.0 \cdot 10^{-15}$	0.69 sec

Pitfalls of BVP approach

Pitfalls of BVP approach

- Require turning asymptotic expansion at one endpoint into asymptotic endpoint at other point. Not easy!

Pitfalls of BVP approach

- Require turning asymptotic expansion at one endpoint into asymptotic endpoint at other point. Not easy!
- Selecting appropriate a_+ and b_- along with indices of truncation is a bit of a black art.

Pitfalls of BVP approach

- Require turning asymptotic expansion at one endpoint into asymptotic endpoint at other point. Not easy!
- Selecting appropriate a_+ and b_- along with indices of truncation is a bit of a black art.
- Actually solving BVP requires choosing starting values for Newton iteration, discretizing the DE, choosing a good step size, etc.

Pitfalls of BVP approach

- Require turning asymptotic expansion at one endpoint into asymptotic endpoint at other point. Not easy!
- Selecting appropriate a_+ and b_- along with indices of truncation is a bit of a black art.
- Actually solving BVP requires choosing starting values for Newton iteration, discretizing the DE, choosing a good step size, etc.

Punchline: BVP approach is insufficiently "black-box" for us.

Better Approach: Numerical Evaluation of Fredholm Determinants

Better Approach: Numerical Evaluation of Fredholm Determinants

Choose your favorite quadrature rule (Clenshaw-Curtis is good) over nodes $x_j \in (a, b)$ and positive weights w_j: $\sum_{j=1}^{m} w_j f(x_j) \approx \int_a^b f(x)dx$

Better Approach: Numerical Evaluation of Fredholm Determinants

Choose your favorite quadrature rule (Clenshaw-Curtis is good) over nodes $x_j \in (a, b)$ and positive weights w_j: $\sum_{j=1}^{m} w_j f(x_j) \approx \int_a^b f(x)dx$
The Fredholm determinant

$$d(z) = \det\left(I - zK \restriction_{L^2_{(a,b)}}\right)$$

has the approximation

$$A_m = K(x_i, y_j)_{i,j=1}^{m}$$

$$d_m(z) = \det\left(\delta_{ij} - z \cdot w_i^{1/2} A_m w_j^{1/2}\right)$$

Evaluating Finite-Dimensional Determinants

Evaluating Finite-Dimensional Determinants

This is a standard Numerical Linear Algebra problem.

Evaluating Finite-Dimensional Determinants

This is a standard Numerical Linear Algebra problem.

- We need the value at a single point $z \in \mathbb{C}$.
  Compute LU of $(I - zA_m)$, get determinant from $\prod_{j=1}^{m} U_{jj}$
  Computing $d_m(z)$ for a single z takes $O(m^3)$ time.

Evaluating Finite-Dimensional Determinants

This is a standard Numerical Linear Algebra problem.

- We need the value at a single point $z \in \mathbb{C}$.
  Compute LU of $(I - zA_m)$, get determinant from $\prod_{j=1}^{m} U_{jj}$
  Computing $d_m(z)$ for a single z takes $O(m^3)$ time.
- We need the value at many points, want $d_m(z)$ as polynomial.
  Compute eigenvalues λ_j of A_m via QR (one-time cost of $O(m^3)$ time, but worse constant factor than LU in practice), then form

$$d_m(z) = \prod_{j=1}^{m} (1 - z\lambda_j)$$

  Computing $d_m(z)$ takes $O(m)$ time.

Sample Matlab Code

The following code computes $F_2(0)$ to one unit of precision in the last decimal place:

```
>> m = 64; [w, x] = ClenshawCurtis(0, inf, m); w2 = sqrt(w);
>> [xi, xj] = ndgrid(x, x);
>> KAi = @AiryKernel;
>> F20 = det(eye(m) - (w2' * w2).*KAi(x, x))
F20 = 0.969372828355262
```

Wrapup

- Computing Fredholm Determinants is faster, easier, and more stable than integrating Painleve IVP or BVP.
- Being able to handle things that are expressed in non-PDE form is useful.
- Bornemann uses the toolset to identify (and subsequently prove) several new results (omitted here for brevity) about distributions of the k-th largest eigenvalue in the soft-edge scaling limit of the GOE and GSE – the numerical code generates immediate insights!