Training Physics-Informed Neural Networks with Preconditioned Gradient-Based Optimizers

Tao Hu

September 2025

1 Introduction

Neural networks have been widely employed to solve partial differential equations(PDEs) especially for those constructed from real-world data[16]. Physics Informed Neural Networks(PINNs) obtain approximate solutions to PDEs by optimizing based on PDE residuals.

Solving PDEs through PINNs has been proven to be a promising area. High-dimensional PDEs can be tackled using deep learning [5], while greedy training algorithms are just sufficient for PINN training [19]. The stability of neural solutions can be ensured through a priori analysis [7]. Different approaches, like Transformers, DOSnet, homotopy dynamics and Newton-informed operators, have been applied to improve the efficiency of PINNs. Transformers can be chosen based on Fourier or Galerkin methods [1], DOSnet offers a non-black-box approach via operator splitting [13], homotopy dynamics can help learning sharp interface solutions [3], and Newton-informed operators enhance nonlinear PDE solving [6].

Nevertheless, it has been demonstrated that the primary problem and difficulty of PINNs is minimizing the residual[11][17]. Minimizing residual loss in PINNs benefits from wide networks and effective activations, and it is suggested that activation functions with bijective k-th order derivatives (for a k-th order PDE) are most effective for minimizing residual loss in PINNs[8].

Various optimizers are applied to minimize the PINN residual. The original PINN framework uses Adam for initial training followed by L-BFGS for final refinement[16]. Dual Cone Gradient Descent addresses multi-objective challenges in PINNs, outperforming SGD and Adam [4], while Implicit SGD stabilizes training for stiff PDEs [9].

A particularly popular area for training PINNs has concentrated on finding other quasi-Newton methods besides L-BFGS. For instances, NysNewton-CG (NNCG) is a second-order method that improves over L-BFGS in loss landscapes [17], and DCGD and ConFIG are designed for multi-objective optimization and

outperform SGD and Adam in fluids [4, 15]. SOAP amazingly achieves gradient alignment in PINN training, which achieves much better efficiency than ADAM in initial training [20].

This work demonstrates that novel optimizers, such as the Dimension-Reduced Second-Order Method (DRSOM)[21] and MUON[10], achieve remarkable performance in training Physics-Informed Neural Networks (PINNs)—a potential that has been previously overlooked.

2 Overview of PINNs

Consider the common partial differential boundary value problem

$$\begin{aligned} \mathcal{D}[u(x), x] &= 0,\ x \in \Omega, \quad (1)\\ \mathcal{B}[u(x), x] &= 0,\ x \in \partial\Omega, \quad (2) \end{aligned}$$

where $\mathcal{D}[u(x), x]$, $\mathcal{B}[u(x), x]$ are smooth functions of $\nabla u(x)$, $u(x)$, and x and $\Omega \subset \mathbb{R}^n$.

Our residual can be represented as

$$\mathcal{L}(\theta) = \underbrace{\frac{1}{N_{bc}} \sum_{i=1}^{N_{bc}} \left|\mathcal{B}[u(x_{bc}^i), x_{bc}^i]\right|^2}_{\text{Data Loss: } \mathcal{L}_{bc}(\theta)} + \underbrace{\frac{1}{N_r} \sum_{i=1}^{N_r} \left|\mathcal{D}[u(x_r^i), x_r^i]\right|^2}_{\text{PDE Loss: } \mathcal{L}_r(\theta)}.$$

In order to solve the boundary value problem using neural network, we need to restrict u to the space of functions that can be represented by neural network.

Hornik et al. [12] proved that, on a compact region, the neural network can approximate not only the value of a differentiable function but also its derivatives. The results can be mainly summarized to the following theorem,

Theorem 2.1. *Assume that $G \in C^m(\mathbb{R})$ and that $G^{(k)} \in \mathcal{L}^1(\mathbb{R})$, $\forall\, 0 \le k \le m$. Assume also that $u \in C^m(\mathbb{R}^n, \mathbb{R})$ and $K \subset \mathbb{R}^n$ is a compact set.*

Define

$$\Sigma(G) = \left\{ \sum_{k=1}^{l} a_i G(\omega_i^T x + b_i) : \omega_i \in \mathbb{R}^n, a_i, b_i \in \mathbb{R}, l \in \mathbb{N} \right\}$$

Then for any $\varepsilon > 0$, there exists $v \in \Sigma(G)$, such that

$$\left|\partial_x^\alpha u(x) - \partial_x^\alpha v\,(x)\right| \le \varepsilon,\ \forall\, |\alpha| \le m,\ x \in K.$$

□

Consider the sigmoid activation function

$$\sigma(x) = \frac{1}{1+e^{-x}}.$$

We can prove that $\sigma(x+1)-\sigma(x-1)$ is in the Schwartz space and thus satisfies the condition of Theorem 2.1 for any $m \in \mathbb{N}$. And noticed that

$$\Sigma(\sigma) \supset \Sigma(\sigma(x+1)-\sigma(x-1)),$$

we conclude that for any $\varepsilon > 0$, there exists $v \in \Sigma(\sigma)$, such that

$$|\partial_x^\alpha u(x) - \partial_x^\alpha v\,(x)| \leq \varepsilon,\ \forall\ |\alpha| \leq m,\ x \in K.$$

For the special case when $m = 1$. Take any $\varepsilon > 0$, there exists $\sigma > 0$ such that $|u(x) - v(x)| < \sigma$ and $\|\nabla u(x) - \nabla v(x)\| < \sigma$, $\forall\ x \in K$, then

$$|\mathcal{D}[u(x), x] - \mathcal{D}[v(x), x]| < \varepsilon \text{ and } |\mathcal{B}[u(x), x] - \mathcal{B}[v(x), x]| < \varepsilon, \forall\ x \in K.$$

Take $u \in C^1(\mathbb{R})$, then $\exists\ v \in \Sigma(\sigma)$, such that $|u(x) - v(x)| < \sigma$ and $\|\nabla u(x) - \nabla v(x)\| < \sigma$. We conclude that

$$|\mathcal{D}[u(x), x] - \mathcal{D}[v(x), x]| < \varepsilon \text{ and } |\mathcal{B}[u(x), x] - \mathcal{B}[v(x), x]| < \varepsilon, \forall\ x \in K.$$

If u is the solution, then

$$\mathcal{D}[u(x), x] = \mathcal{B}[u(x), x] = 0.$$

We conclude that

$$|\mathcal{D}[v(x), x]| < \varepsilon \text{ and } |\mathcal{B}[v(x), x]| < \varepsilon.$$

3 The DRSOM

The update rule for DRSOM is given by:

$$x_{k+1} = x_k - \alpha_1 g_k + \alpha_2 d_k$$

This can be expressed in matrix form as:

$$x_{k+1} = x_k + \begin{pmatrix} -g_k & d_k \end{pmatrix} \begin{pmatrix} \alpha_1 \\ \alpha_2 \end{pmatrix}$$

where:

- $g_k = \nabla \mathcal{L}(x_k)$ is the gradient of the objective function $\mathcal{L}$ at the current iterate x_k.

- $d_k = x_k - x_{k-1}$ is the direction from the previous step.

The coefficient vector $\alpha = (\alpha_1, \alpha_2)^T$ is determined by solving the following trust-region subproblem, which minimizes a quadratic model $m_k(\alpha)$ of the objective function:

$$\alpha = \arg \min_{\|-\alpha_1 g_k + \alpha_2 d_k\| \le \rho_k} m_k(\alpha)$$

The quadratic model is defined as:

$$m_k(\alpha) = \mathcal{L}(x_k) + c_k^T \alpha + \frac{1}{2} \alpha^T Q_k \alpha$$

The components of this model, vector c_k and matrix Q_k, are constructed as follows:

$$c_k = \begin{pmatrix} -\|g_k\|^2 \\ g_k^T d_k \end{pmatrix}, \quad Q_k = \begin{pmatrix} g_k^T H(x_k) g_k & -g_k^T H(x_k) d_k \\ -g_k^T H(x_k) d_k & d_k^T H(x_k) d_k \end{pmatrix}$$

Here, $H(x_k)$ is the Hessian matrix of $\mathcal{L}$ evaluated at x_k, and ρ_k is the trust-region radius, which is adjusted at each iteration according to a specific rule.

Two different attempts have been used to calculate the quadratic forms $g_k^T H(x_k) g_k$, $g_k^T H(x_k) d_k$, and $d_k^T H(x_k) d_k$:

- Automatic differentiation(AD): Obtaining $H(x_k)v$ by doing backward propagation on $\nabla \mathcal{L}(x_k)^T v$;
- Finite difference(FD): A typical example would be approximating $H(x_k)v$ by $\frac{1}{\delta} (\nabla \mathcal{L}(x_k + \delta v) - \nabla \mathcal{L}(x_k))$.

Although it has been demonstrated that automatic differentiation is essential in PINN training for the reason that finite difference makes tiny eigenvalues larger, leading to inexactness when calculating the pseudo-inverse. However, automatic differentiation (AD) can sometimes lead to memory overflow issues due to the storage requirements of the computational graph[2].

3.1 The straightforward calculation of two Hessian–vector products in DRSOM

Let $g_k := \nabla \mathcal{L}(x_k)$ and $H_k := \nabla^2 \mathcal{L}(x_k)$. On a compact region $\mathcal{R}$ containing all iterates and trial points, assume

$$\|\nabla \mathcal{L}(x)\| \le G, \quad |\mathcal{L}(x)| \le M, \quad \|\nabla^2 \mathcal{L}(x)\| \le L_1, \quad \|\nabla^3 \mathcal{L}(x)\| \le L_2, \quad \|\nabla^4 \mathcal{L}(x)\| \le L_3, \qquad \forall x \in \mathcal{R}.$$

By Taylor's theorem for the gradient, for any $\delta > 0$,

$$\nabla \mathcal{L}(x_k \pm \delta g_k) = g_k \pm \delta H_k g_k + r_{\pm,k}, \qquad \|r_{\pm,k}\| \le \tfrac{1}{2} L_2 \, \delta^2 \|g_k\|^2.$$

Hence

$$H_k g_k \;\approx\; \frac{g_k - \nabla \mathcal{L}(x_k - \delta \, g_k)}{\delta}, \qquad \left\| H_k g_k - \frac{g_k - \nabla \mathcal{L}(x_k - \delta \, g_k)}{\delta} \right\| \le \tfrac{1}{2} L_2 \, \delta \, \|g_k\|^2.$$

For the second product, let $s_{k-1} := x_k - x_{k-1}$ (your d_{k-1}). Then

$$g_k - g_{k-1} = \left(\int_0^1 \nabla^2 \mathcal{L}(x_{k-1} + t s_{k-1})\, dt \right) s_{k-1},$$

so that

$$H(x_{k-1}) s_{k-1} \;\approx\; g_k - g_{k-1}, \qquad \left\| (g_k - g_{k-1}) - H(x_{k-1}) s_{k-1} \right\| \le \tfrac{1}{2} L_2 \, \|s_{k-1}\|^2.$$

Remark. If one needs $H_k d_k$, an analogous relation is $H_k d_k \approx g_{k+1} - g_k$ with the same $O(\|d_k\|^2)$ error.

3.2 Regularizing an ill-conditioned Q_k

Let $Q_k \in \mathbb{R}^{2\times 2}$ be symmetric with eigenvalues $\lambda_{\min} \le \lambda_{\max}$. For a target $\tau > 1$, the smallest shift $\varepsilon \ge 0$ such that $Q_k + \varepsilon I_2 \succ 0$ and

$$\kappa_2(Q_k + \varepsilon I_2) = \frac{\lambda_{\max} + \varepsilon}{\lambda_{\min} + \varepsilon} \le \tau$$

is

$$\varepsilon^\star = \max\left\{ 0,\; -\lambda_{\min} + \sigma,\; \frac{\lambda_{\max} - \tau\, \lambda_{\min}}{\tau - 1} \right\},$$

with a tiny safety margin $\sigma > 0$ (e.g. 10^{-12}). Since $\kappa_2(Q + \varepsilon I)$ decreases monotonically in ε, this choice is minimal.

3.3 Reducing the number of gradient evaluations

At late stages, $\|d_{k-1}\|$ is small, and

$$\begin{cases} H_k\, g_k \;\approx\; \dfrac{g_k - \nabla \mathcal{L}(x_k - \delta\, g_k)}{\delta}, \\[2ex] H(x_{k-1})\, d_{k-1} \;\approx\; g_k - g_{k-1}, \end{cases} \tag{3}$$

which suffices to assemble Q_k with only one extra gradient evaluation at $x_k - \delta g_k$.

3.4 Error analysis (finite differences and round-off)

Define

$$\phi_1(\delta) := \frac{\|g_k\|^2 - g_k^\top \nabla \mathcal{L}(x_k - \delta\, g_k)}{\delta}. \tag{4}$$

Then

$$\left| g_k^\top H_k g_k - \phi_1(\delta) \right| \le \tfrac{1}{2} L_2\, \delta\, \|g_k\|^3,$$

and, under the standard floating-point model with inner-product error γ_n,

$$\left| \phi_1(\delta) - \mathrm{fl}(\phi_1(\delta)) \right| \le \frac{C_1\, \gamma_n\, \|g_k\|^2}{\delta} \quad (C_1 \in [2, 4]).$$

Balancing gives $\delta_1^\star \approx \sqrt{(2C_1\gamma_n)/(L_2\|g_k\|)}$.

3.5 Hermite approximation (higher accuracy with function & gradient)

Using the expansions of $\mathcal{L}$ and $\nabla\mathcal{L}$ at x_k along g_k and canceling cubic terms,

$$g_k^\top H_k g_k = \frac{\mathcal{L}(x_k - \delta g_k) - \mathcal{L}(x_k) + \frac{\delta}{3} g_k^\top \nabla\mathcal{L}(x_k - \delta g_k) + \frac{2\delta}{3}\|g_k\|^2}{\delta^2/6} + \mathcal{O}(\delta^2\|g_k\|^4), \tag{5}$$

with the explicit bound

$$\left| g_k^\top H_k g_k - \frac{\mathcal{L}(x_k - \delta g_k) - \mathcal{L}(x_k) + \frac{\delta}{3} g_k^\top \nabla\mathcal{L}(x_k - \delta g_k) + \frac{2\delta}{3}\|g_k\|^2}{\delta^2/6} \right| \leq \frac{7}{12} L_3\, \delta^2\, \|g_k\|^4.$$

In floating-point arithmetic, a practical choice is

$$\delta_H^\star \approx \Big(\frac{C_f\, u\, M}{L_3\, \|g_k\|^4}\Big)^{1/4}.$$

4 The MUON

Muon is also an optimizer designed for the layer-wise structure of the neural network[10]. Assuming that $G_t \in \mathbb{R}^{m\times n}, m \geq n$, the updating formula is given as Algorithm 1.

Algorithm 1 Muon

Require: Learning rate η, momentum μ
1: Initialize $B_0 \leftarrow 0$
2: **for** $t = 1, \ldots$ **do do**
3: Compute gradient $G_t \leftarrow \nabla\mathcal{L}(\theta_t)$
4: $B_t \leftarrow \mu B_{t-1} + G_t$
5: $O_t \leftarrow \textsc{NewtonSchulz5}(B_t)$
6: Update parameters $\theta_t \leftarrow \theta_{t-1} - \eta O_t$
7: **end for**
8: **return** θ_t

while NewtonSchulz5 is an approximation to the following mapping:

$$\begin{array}{ccc} \mathbb{R}^{m\times n} & \to & \mathbb{R}^{m\times n} \\ U^T\Lambda V & \mapsto & U^T V. \end{array}$$

This approximation is done by the composition of simpler functions,

$$\text{NewtonSchulz5} = \varphi \circ \varphi \circ \varphi \circ \varphi \circ \varphi.$$

$\varphi(U^T\Lambda V) = U^T(a\Lambda + b\Lambda^3 + c\Lambda^5)V.$

This approximation makes the computational complexity from $\mathcal{O}(m^3)$ to $\mathcal{O}(m^2)$.

Liu et. al. [14] shows that in the task of LLM training, the MUON optimizer performs better with the weight decay mechanism:

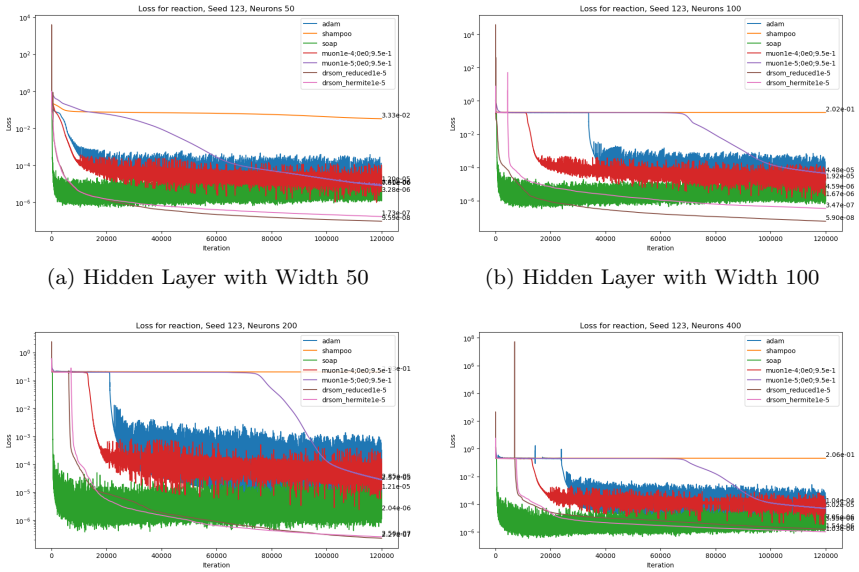
$$W_t = W_{t-1} - \eta_t(O_t + \lambda W_{t-1}).$$

An interesting note is that MUON with weight-decay can be understood as a kind of Stochastic Frank-Wolfe[18].

5 Numerical Comparison between ADAM, SOAP, MUON and DRSOM

From Figure 1, 2,3 we can observe the following patterns:

1. The optimizers varies greatly from different problems,
2. The SOAP optimizer always excels in the early stages,
3. The SOAP optimizer always performs bad in the late stages,
4. The DRSOM optimizer often excels in the late stages.



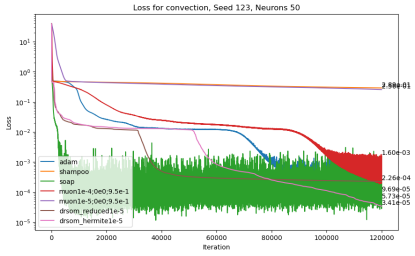
(a) Hidden Layer with Width 50

(b) Hidden Layer with Width 100

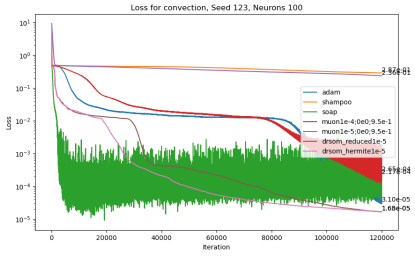
(c) Hidden Layer with Width 200

(d) Hidden Layer with Width 400

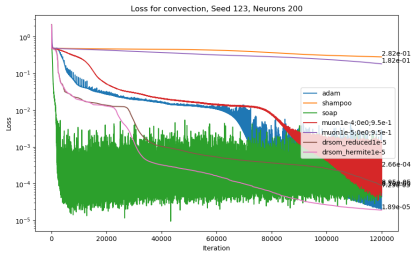
Figure 1: Comparison between ADAM, SOAP, MUON and DRSOM for logistic equations



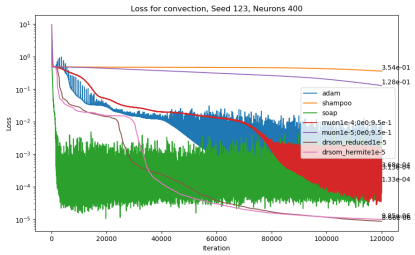
(a) Hidden Layer with Width 50



(b) Hidden Layer with Width 100



(c) Hidden Layer with Width 200



(d) Hidden Layer with Width 400

Figure 2: Comparison between ADAM, SOAP, MUON and DRSOM for transport equations

6 Tackling the Disadvantage of MUON in PINN training-Using a Proper Step Size Schedule

In Algorithm 1, instead of using a constant step size η, we replace η by $\eta_t = \kappa\mathcal{L}(\theta_t)$. A nice finding is that this step-size schedule with $\kappa = 1$ successfully solves the Logistic equation with high accuracy.

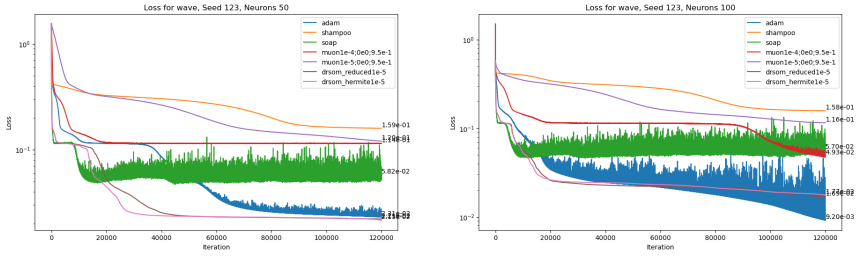
7 Future Works

While it is very interesting that such a simple step-size schedule would resolve the zig-zag pattern and achieve a strong decrease in the final refinement, the effect on other PDEs remains unclear and remains to be an interesting topic.

References

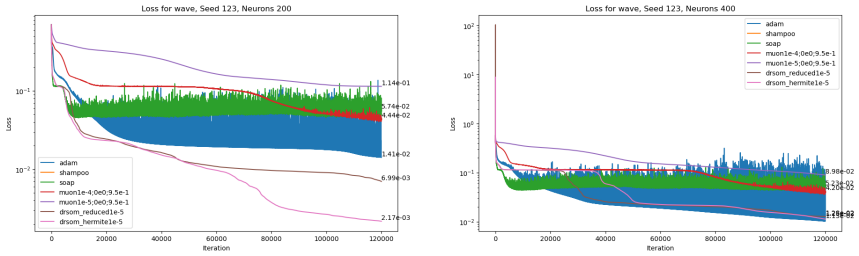
[1] S. Cao. Choose a transformer: fourier or galerkin. In *Advances in Neural Information Processing Systems*, volume 34, pages 24924–24940, 2021.

[2] C. Chen, Y. Yang, Y. Xiang, and W. Hao. Automatic differentiation is essential in training neural networks for solving differential equations. *J Sci Comput*, 104(54), 2025.



(a) Hidden Layer with Width 50

(b) Hidden Layer with Width 100



(c) Hidden Layer with Width 200

(d) Hidden Layer with Width 400

Figure 3: Comparison between ADAM, SOAP, MUON and DRSOM for wave equations

[3] C. Chen, Y. Yang, Y. Xiang, and W. Hao. Learn sharp interface solution by homotopy dynamics, 2025.

[4] Y. Du, H. Son, D. Lim, N. Zhang, and T. Zaki. Dual cone gradient descent for training physics-informed neural networks. In *Advances in Neural Information Processing Systems*, 2024.

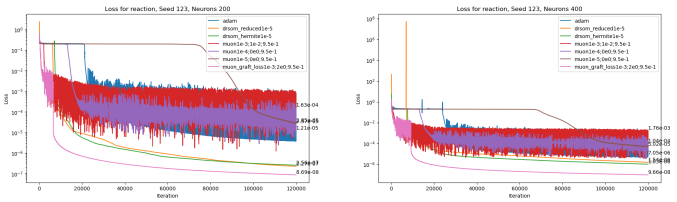
[5] J. Han, A. Jentzen, and W. E. Solving high-dimensional partial differential equations using deep learning. *Proceedings of the National Academy of Sciences*, 115(34):8505–8510, 2018.

[6] W. Hao, R. Li, Y. Xi, T. Xu, and Y. Yang. Multiscale neural networks for approximating green's functions, 2024.

[7] Q. Hong, J. Siegel, and J. Xu. A priori analysis of stable neural network solutions to numerical pdes, 2021.

[8] N. Hosseini Dashtbayaz, G. Farhani, B. Wang, and C. X. Ling. Physics-informed neural networks: Minimizing residual loss with wide networks and effective activations, 2024.

[9] J. Jin, M. Mattheakis, P. Protopapas, and M. Tegmark. Implicit stochastic gradient descent for training physics-informed neural networks, 2023.



(a) Hidden Layer with Width 200

(b) Hidden Layer with Width 400

Figure 4: Comparison between different variants of MUON

[10] K. Jordan, Y. Jin, V. Boza, J. You, F. Cesista, L. Newhouse, and J. Bernstein. Muon: An optimizer for hidden layers in neural networks. https://kellerjordan.github.io/posts/muon/, 2024. Accessed: September 23, 2025.

[11] A. S. Krishnapriyan, A. Gholami, S. Zhe, R. M. Kirby, and M. W. Mahoney. Characterizing possible failure modes in physics-informed neural networks, 2021.

[12] H. W. Kurt Hornik, Maxwell Stinchcombe. Approximation of an unknown mapping and its derivatives using multilayer feedforward networks. *Neural Networks*, 3:551–560, 1990.

[13] Y. Lan, Z. Li, J. Sun, and Y. Xiang. Dosnet as a non-black-box pde solver: When deep learning meets operator splitting. *Journal of Computational Physics*, 491:112343, 2023.

[14] J. Liu, J. Su, X. Yao, Z. Jiang, G. Lai, Y. Du, Y. Qin, W. Xu, E. Lu, J. Yan, et al. Muon is scalable for llm training. *arXiv preprint arXiv:2502.16982*, 2025.

[15] Q. Liu, J. Rothfuss, B. Schutter, I. Redko, and M. Jaggi. Config: Towards conflict-free training of physics informed neural networks, 2024. ICLR 2025.

[16] M. Raissi, P. Perdikaris, and G. Karniadakis. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *Journal of Computational Physics*, 378:686–707, 2019.

[17] P. Rathore, W. Lei, Z. Frangella, L. Lu, and M. Udell. Challenges in training pinns: A loss landscape perspective, 2024.

[18] M.-E. Sfyraki. Lions and muons: Optimization via stochastic frank-wolfe, 2025.

[19] J. Siegel, Q. Hong, X. Jin, W. Hao, and J. Xu. Greedy training algorithms for neural networks and applications to pdes. *Journal of Computational Physics*, 484:112084, 2023.

[20] S. Wang, Y. Chen, B. Li, and P. Perdikaris. Gradient alignment in physics-informed neural networks: A second-order optimization perspective, 2025.

[21] C. Zhang. Drsom: A dimension reduced second-order method, 2022.

Study Reports: Notes on Randomized Sketching and Krylov Spaces

Tao Hu[1]

[1] School of Mathematics and Statistics, Xi'an Jiaotong University, Xi'an

tao_hu2358@outlook.com

1. Why randomization?

Randomization is a promising tool to develop faster Numerical Linear Algebra Solvers.

To deal with large sparse matrices, we can use preconditioning and BLAS-level.

2. Randomized Preconditioning

Solving an over-determined least–squares system

$$\min_{x\in\mathbb{R}^m} \|Ax-b\|_2, \qquad A\in\mathbb{R}^{N\times m},\ N\gg m, \tag{1}$$

with a naive iterative method (e.g. LSQR) can be slow when A is ill-conditioned. *Randomized preconditioning* uses a cheap dimension-reducing map to build a near-optimal right-preconditioner in $O(\mathrm{nnz}(A))$ time where $\mathrm{nnz}(A)$ refers to the number of nonzero elements of A, a measurement of the sparsity of A.

2.1. LSQR method

LSQR is an iterative Krylov method. In exact arithmetic its iterates are identical to those produced by Conjugate Gradient applied to the normal equations $A^{\mathsf{T}}A\,x = A^{\mathsf{T}}b$ (often called *CGLS* or *CGNR*), but LSQR achieves this through Golub–Kahan bidiagonalisation, which is numerically more robust.

Step 1: Initialisation (k = 1)(paige1982lsqr)(scipyLSQR)(matlabLSQR)

$$\beta_1 = \|b\|_2, \quad u_1 = b/\beta_1, \quad \alpha_1 = \|A^{\mathsf{T}}u_1\|_2, \quad v_1 = A^{\mathsf{T}}u_1/\alpha_1, \quad w_1 = v_1, \quad x_0 = 0.$$

Step 2: Golub–Kahan bidiagonal step (k = 1,2,...)

$$u_{k+1} = Av_k - \alpha_k u_k, \qquad \beta_{k+1} = \|u_{k+1}\|_2,\ u_{k+1} \leftarrow u_{k+1}/\beta_{k+1},$$
$$v_{k+1} = A^{\mathsf{T}}u_{k+1} - \beta_{k+1}v_k, \quad \alpha_{k+1} = \|v_{k+1}\|_2,\ v_{k+1} \leftarrow v_{k+1}/\alpha_{k+1}.$$

Step 3: Apply the Givens rotation (plane rotation)

$$\rho_k = \sqrt{\rho_{k-1}^2+\beta_{k+1}^2}, \quad c_k = \rho_{k-1}/\rho_k, \quad s_k = \beta_{k+1}/\rho_k,$$
$$\theta_{k+1} = s_k\,\alpha_{k+1}, \qquad \rho_k = -c_k\,\alpha_{k+1},$$
$$\varphi_k = c_k\,\varphi_{k-1}, \qquad \varphi_{k+1} = s_k\,\varphi_{k-1},$$

where $\rho_0 := \alpha_1$ and $\varphi_0 := \beta_1\alpha_1$.

Step 4: Solution and direction updates

$$x_k = x_{k-1} + \frac{\varphi_k}{\rho_k}\,w_k, \qquad w_{k+1} = v_{k+1} - \frac{\theta_{k+1}}{\rho_k}\,w_k.$$

The residual norm satisfies $\|r_k\|_2 = \varphi_{k+1}$, so it decreases monotonically.

2.2. Blendenpik's method

Assume $A \in \mathbb{R}^{N\times m}$ with $N \gg m$ and consider the over- determined least–squares problem $\min_x \|Ax - b\|_2$. Let the thin QR of A be $A = Q_A R_A$, where $Q_A \in \mathbb{R}^{N\times m}$ has orthonormal columns and $R_A \in \mathbb{R}^{m\times m}$ is upper triangular. The *optimal* (but expensive) right-preconditioner is R_A^{-1}. Blendenpik replaces it by a cheap approximation obtained from a *sketch*.

Sketch step. Draw a Johnson–Lindenstrauss matrix $S \in \mathbb{R}^{s\times N}$, e.g. an SRHT with $s = \lceil 4m \log m \rceil$ rows. Form[1]

$$B \;=\; SA \;=\; SQ_A R_A.$$

Preconditioner construction. Compute the thin QR of B: $B = Q_B R_B$ with $Q_B^{\mathsf{T}} Q_B = I_m$. Set the right-preconditioner

$$P \;:=\; R_B^{-1}.$$

Theorem 1 (Avron–Maymounkov–Toledo, 2010)**.** *With the notation above,*

$$\kappa\big(AP\big) \;=\; \kappa\big(SQ_A\big).$$

If S is an ε-subspace embedding for $\operatorname{range}(A)$*, then* $\kappa(AP) \;\le\; \sqrt{\dfrac{1+\varepsilon}{1-\varepsilon}}$.

Demonstração. Write $P = R_B^{-1}$. Because $A = Q_A R_A$ and $SA = Q_B R_B$, we have $SQ_A R_A = Q_B R_B$, hence $SQ_A = Q_B R_B R_A^{-1}$. Left-multiplying by an orthonormal matrix does not change singular values, so $\kappa(SQ_A) = \kappa(R_B R_A^{-1})$. Similarly, $AP = Q_A R_A R_B^{-1}$ and therefore $\kappa(AP) = \kappa(R_A R_B^{-1})$. Since $\kappa(M) = \kappa(M^{-1})$ for any invertible M,

$$\kappa(R_B R_A^{-1}) = \kappa\big((R_A R_B^{-1})^{-1}\big) = \kappa(R_A R_B^{-1}),$$

which shows $\kappa(AP) = \kappa(SQ_A)$. For a Johnson–Lindenstrauss sketch the latter is bounded by $\sqrt{\frac{1+\varepsilon}{1-\varepsilon}}$, completing the proof. □

Blendenpik algorithm.

Step 1: Draw S (SRHT) and compute $B = SA$.
Step 2: QR-factorise B in BLAS-3, obtain R_B^{-1}.
Step 3: Run LSQR on $APy = b$; set $x = Py$.

Typical settings $s \approx 2m$–$4m$ yield $\kappa(AP) \lesssim 4$ and LSQR converges in $\mathcal{O}(m)$ operations; the total flop count is $O(\operatorname{nnz}(A)\log s + m^3)$. On dense inputs this is 5–10× faster than LAPACK's QR with identical backward error.

[1] The product SA costs $O(\operatorname{nnz}(A)\log s)$ flops with an SRHT.

2.3. Johnson–Lindenstrauss Lemma (1984)

Theorem 2 (Johnson–Lindenstrauss)**.** *Let $0 < \varepsilon < 1$ and let $X \subset \mathbb{R}^N$ be a set of m points. If*

$$s \;\geq\; \frac{8 \log m}{\varepsilon^2},$$

then there exists a linear map $f : \mathbb{R}^N \to \mathbb{R}^s$ such that for all $u, v \in X$

$$(1-\varepsilon)\,\|u-v\|_2^2 \;\leq\; \|f(u)-f(v)\|_2^2 \;\leq\; (1+\varepsilon)\,\|u-v\|_2^2.$$

2.4. Subspace Embeddings

Definition 1. *Let $V \subset \mathbb{R}^N$ be an m-dimensional subspace and let $\varepsilon \in (0,1)$. A matrix $S \in \mathbb{R}^{s \times N}$ is an* ε-subspace embedding *for V if*

$$(1-\varepsilon)\,\|v\|^2 \;\leq\; \|Sv\|^2 \;\leq\; (1+\varepsilon)\,\|v\|^2, \quad \forall v \in V.$$

The definition is equivalent to

$$u^\top v \;-\; \varepsilon\,\|u\|\,\|v\| \;\leq\; (Su)^\top (Sv) \;\leq\; u^\top v \;+\; \varepsilon\,\|u\|\,\|v\|, \quad \forall u, v \in V.$$

For many random constructions one can choose

$$s \;=\; \Theta\big((m + \ln(1/\delta))\,\varepsilon^{-2}\big),$$

so that the embedding holds with probability at least $1-\delta$. (The constant hidden in $\Theta(\cdot)$ is typically between 4 and 8.) :contentReference[oaicite:0]index=0

Definition 2. *Let Π be a distribution over $s \times N$ matrices S (where s may depend on $N, m, \varepsilon, \delta$). If, with probability at least $1-\delta$, a draw $S \sim \Pi$ is an ε-subspace embedding for* every *fixed m-dimensional subspace V, we call Π an $(N, m, \varepsilon, \delta)$* OSE ensemble.

Theorem 3. *The ensemble of $s \times N$ matrices whose entries are i.i.d. $\mathcal{N}(0, s^{-1})$ is an $(N, m, \varepsilon, \delta)$ OSE provided*

$$s \;=\; O\big((m + \ln(1/\delta))\,\varepsilon^{-2}\big).$$

Idea. For any unit vector $x \in \mathbb{R}^N$, $(Sx)_i \sim \mathcal{N}(0, s^{-1})$, so $s\,\|Sx\|^2 \sim \chi^2_s$. Using the tail bounds $\Pr\big[\chi^2_s \geq (1+\varepsilon)s\big] \leq e^{-\varepsilon^2 s/4}$ and $\Pr\big[\chi^2_s \leq (1-\varepsilon)s\big] \leq e^{-\varepsilon^2 s/4}$, together with a union bound over an $\varepsilon/2$-net of the unit sphere in V (whose size is at most $(1+4/\varepsilon)^m$), yields the stated dimension requirement. $\square$

Common OSE constructions.

- **Gaussian or Rademacher** (dense) matrices – optimal dimension, $O(Ns)$ multiply.
- **SRHT / SRFT** (structured) – $O(N \log N)$ multiply, same dimension up to log factors.
- **CountSketch / TensorSketch** (sparse) – $O(\mathrm{nnz}(A))$ multiply; trade-offs between sparsity and s.

Lower-bound results show that the scaling $s = \Omega((m+\ln(1/\delta))\,\varepsilon^{-2})$ is information-theoretically optimal, even for sparse embeddings.

2.5. $N \log(s)$ order of computations

For a sparsity parameter ζ, construct

$$S \;=\; \sqrt{\frac{N}{\zeta}}\,[\,s_1, s_2, \ldots, s_N\,], \;\; \text{where } [\,s_1, s_2, \ldots, s_N\,] \;\in\; \{-1, 0, 1\}^{s\times N}.$$

For each column $j \in [N]$:

- pick *distinct* row indices $\rho_{j,1}, \ldots, \rho_{j,\zeta} \in [s]$ (sampling *without replacement*, so no two coincide);
- pick signs $\sigma_{j,1}, \ldots, \sigma_{j,\zeta} \in \{-1, 1\}$.

Set $(s_j)_i = \sigma_{j,\ell}$ if $i = \rho_{j,\ell}$ for some ℓ and $(s_j)_i = 0$ otherwise. Each column therefore has exactly ζ non-zeros.

Fast transforms (e.g. FFT/FWHT for SRHT or simple hashing for CountSketch) reduce the naive $O(sN)$ multiplication cost to $O\big(N \log s\big)$ for dense data or $O\big(\zeta\ \mathrm{nnz}(A)\big)$ for sparse data.

2.6. Krylov Methods

The Arnoldi process is the workhorse of many Krylov solvers, e.g. GMRES, EIGS.

Review: Gram-Schmidt Process

Algorithm 1 Modified Gram–Schmidt step inside Arnoldi

for $j = 1, \ldots, m$ **do**
  $w \leftarrow A v_j$ ▷ new Krylov vector
  for $i = 1, \ldots, j$ **do** ▷ inner projection loop
    $h_{ij} \leftarrow v_i^\top w$
    $w \leftarrow w - h_{ij}\, v_i$
  end for
  $h_{j+1,j} \leftarrow \|w\|_2$
  $v_{j+1} \leftarrow w / h_{j+1,j}$
end for

Gram-Schmidt process for orthonormal basis (for the $m-$ th Krylov space for $A \in \mathbb{R}^{N\times N}$.

$$V_m = [v_1, v_2, \ldots, v_m].$$

The complexity for such Gram-Schmidt process is $O(Nm^2)$.

2.7. Krylov Method

Given a matrix $A \in \mathbb{C}^{N\times N}$ and a non-zero vector $b \in \mathbb{C}^N$, the m-th *Krylov subspace* is

$$\mathcal{K}_m(A, b) \;=\; \mathrm{span}\big\{\, b,\; Ab,\; A^2 b, \ldots, A^{m-1} b \big\}.$$

The **Arnoldi process** builds an orthonormal basis $V_m = [\,v_1, \ldots, v_m\,]$ of $\mathcal{K}_m(A, b)$ and a small upper-Hessenberg matrix H_m such that

$$A\,V_m \;=\; V_m H_m \;+\; h_{m+1,m}\, v_{m+1} e_m^*. \tag{1}$$

Appending the extra vector and entry gives the compact form

$$A\,V_m \;=\; V_{m+1}\,\underline{H}_m, \qquad \underline{H}_m = \begin{bmatrix} H_m \\ h_{m+1,m}\,e_m^* \end{bmatrix}. \tag{2}$$

Algorithm 2 Arnoldi (MGS)

```
v_1 ← b/‖b‖_2
for j = 1, ..., m do
    w ← A v_j                                   ▷ new Krylov vector
    for i = 1, ..., j do                        ▷ orthogonalise
        h_ij ← v_i^* w
        w ← w − h_ij v_i
    end for
    h_{j+1,j} ← ‖w‖_2                           ▷ may trigger re-orthogonalisation
    v_{j+1} ← w/h_{j+1,j}
end for
```

Arnoldi iteration with modified Gram–Schmidt

- **Work**.$O(Nm^2)$ flops for dense A; $O(\mathrm{nnz}(A)\,m + Nm^2)$ when A is sparse.
- **Storage**.$O(Nm)$ for V_{m+1} and $O(m^2)$ for $\underline{H}_m$.
- **Stability**.Modified GS (or classical GS with re-orthogonalisation) keeps the columns of V_m nearly orthonormal.

Projection methods built on (2)

FOM. Impose $V_m^* r_m = 0$ with $x_m = x_0 + V_m y_m$.Solve $H_m y_m = \beta e_1$.
GMRES. Minimise $\|r_m\|_2 = \|\beta e_1 - \underline{H}_m y\|_2$ to obtain $x_m = x_0 + V_m y_m$.
EIGS / IRAM. The eigenvalues of H_m (*Ritz values*) approximate some eigenvalues of A.

Complexity versus accuracy

- **Restarting** (e.g. GMRES(k)) caps the subspace dimension at k, keeping the cost per cycle $O(Nk^2)$.
- **Selective re-orthogonalisation** orthogonalises only against "dangerous" directions to reduce the inner loop.
- **Block Arnoldi** groups p right-hand sides and replaces dot-products by level-3 BLAS, improving cache efficiency.

The Arnoldi relation and its associated small Hessenberg matrix $\underline{H}_m$ therefore provide the foundation for residual- minimising linear solvers (GMRES/FOM) and large-scale eigenvalue codes (ARPACK, MATLAB's `eigs`), balancing $O(Nm^2)$ orthogonalisation cost against rapid convergence.

2.7.1. FOM and GMRES approximants

Once the Arnoldi relation is available, two standard Petrov–Galerkin projections arise:

$$x_m^{\text{FOM}} \;=\; V_m\, H_m^{-1}\big(V_m^* b\big), \qquad x_m^{\text{GMRES}} \;=\; V_m\, \underline{H}_m^{\dagger}\big(V_{m+1}^* b\big),$$

where $\underline{H}_m^{\dagger}$ denotes the minimum–norm $(m+1)\times m$ pseudoinverse obtained, e.g., from the thin QR of $\underline{H}_m$.

Residual properties

$$\begin{aligned} r_m^{\text{FOM}} &= b - Ax_m^{\text{FOM}} && \perp\ \mathcal{K}_m(A,b), \\ r_m^{\text{GMRES}} &= b - Ax_m^{\text{GMRES}} && \text{minimises } \|r_m\|_2 \text{ over } x_0 + \mathcal{K}_m(A,b). \end{aligned}$$

How the formulas arise

- **FOM.** Impose the Petrov–Galerkin condition $V_m^* r_m = 0$: $V_m^* b = V_m^* A V_m y$. Since $V_m^* A V_m = H_m$, $y = H_m^{-1} V_m^* b$ and $x_m = V_m y$.
- **GMRES.** Write $r_m = b - AV_m y$. Using (**??**), $\|r_m\|_2 = \|\beta e_1 - \underline{H}_m y\|_2$. Minimising this $(m+1)\times m$ least squares gives $y = \underline{H}_m^{\dagger}(\beta e_1)$, hence the expression above.

Cost summary

Task	Dense A	Sparse A
Arnoldi (m steps)	$O(Nm^2)$	$O(\mathrm{nnz}(A)m + Nm^2)$
Solve $H_m y = \beta e_1$ (FOM)	$O(m^2)$	$O(m^2)$
Solve $\underline{H}_m y \approx \beta e_1$ (GMRES)	$O(m^2)$	$O(m^2)$
Form $x_m = V_m y$	$O(Nm)$	$O(Nm)$

Thus forming either x_m^{FOM} or x_m^{GMRES} after the basis is built costs $O(Nm + m^2)$ extra operations and negligible additional storage.

Remarks

- For Hermitian A, H_m is tridiagonal and the cost per step drops to $O(Nm)$.
- In practice GMRES is restarted after $m = k$ iterations to keep the $O(Nk^2)$ growth of orthogonalisation under control; FOM is rarely used without restarting.
- The residual norm in GMRES can be updated cheaply from the Givens rotations already applied while solving the least–squares problem, giving a robust stopping criterion.

2.8. Sketch the Arnoldi Process (RGS–Arnoldi)

Instead of computing inner products in full precision, replace the standard Gram–Schmidt projection $\langle u, v\rangle$ with the *sketched* inner product $\langle Su, Sv\rangle$, where $S \in \mathbb{R}^{s\times N}$ is a Johnson–Lindenstrauss embedding with distortion ε.

The arithmetic cost remains $O(Nm^2)$, but the algorithm now produces an orthonormal basis SV_m *and* a well-conditioned unsketched basis V_m satisfying

$$\left(\frac{1-\varepsilon}{1+\varepsilon}\right)^{1/2} \kappa(SV_m) \;\le\; \kappa(V_m) \;\le\; \left(\frac{1+\varepsilon}{1-\varepsilon}\right)^{1/2} \kappa(SV_m),$$

where $\kappa(\cdot)$ denotes the 2-norm condition number.

Recall. Classical GMRES chooses the optimal iterate $x_m \in \mathcal{K}_m(A, b)$ by

$$x_m \;=\; \operatorname*{arg\,min}_{x \in \mathcal{K}_m(A,b)} \| b - Ax \|_2.$$

Writing $x_m = V_m y_m$ gives $y_m = (AV_m)^\dagger b$.

Sketched GMRES. Replace the normal equations by their sketched analogue:

$$\hat{y}_m \;=\; \big(SAV_m\big)^\dagger Sb, \qquad \hat{x}_m \;=\; V_m \hat{y}_m.$$

Thus

$$x_m = \operatorname*{arg\,min}_{x \in V_m \mathbb{C}^m} \| b - Ax \|_2, \qquad \hat{x}_m = \operatorname*{arg\,min}_{x \in V_m \mathbb{C}^m} \| Sb - SAx \|_2.$$

Residual bounds. With $\|r_m\| = \|b - Ax_m\|$ and $\hat{r}_m = b - A\hat{x}_m$, the sketch preserves the GMRES residual up to

$$\|r_m\| \;\le\; \|\hat{r}_m\| \;\le\; \frac{1}{\sqrt{1-\varepsilon}} \|S\hat{r}_m\| \;\le\; \frac{1}{\sqrt{1-\varepsilon}} \|Sr_m\| \;\le\; \sqrt{\frac{1+\varepsilon}{1-\varepsilon}}\, \|r_m\|.$$

Forming either x_m or $\hat{x}_m$ after the basis V_m is built costs $O(Nm + m^2)$ additional operations, so the overall complexity of sketched Arnoldi remains dominated by the orthogonalisation phase.

2.9. Sketched Deflated Restarting (GMRES–SDR)

Sketched GMRES gains speed but its error bounds contain *extra* condition–number factors—one from the sketch $\kappa(SV_k)$ and one from the deflation subspace $\kappa(SU)$. GMRES–SDR controls these by

1. keeping the sketch dimension $s = \Theta\big((k + \ln(1/\delta))\, \varepsilon^{-2}\big)$;
2. re-orthogonalising the recycled Ritz vectors after each cycle;
3. scaling the preconditioner so that $\kappa(SU) \le (1 + \varepsilon)$.

With those safeguards the usual GMRES residual bound $\|r_m\| \le (1+\varepsilon) \min_{p \in \Pi_m} \|p(A) r_0\|$ remains valid and the per-cycle cost stays $O(\mathrm{nnz}(A)k + Nk^2)$.

2.10. Stabilise sGMRES through FGMRES

We treat *sketched* GMRES (sGMRES) as a *right-preconditioner* inside Flexible GMRES (FGMRES).

Theorem 4. *Let r_j^{FGMRES} be the residual after j outer iterations of FGMRES and let $r_j^{PREC} = Az_j - v_j$ be the inner (preconditioner) residual returned by the j-th call to sGMRES. Then*

$$\|r_j^{FGMRES}\| \;\le\; \|r_{j-1}^{FOM}\|\,\|r_j^{PREC}\|,$$

where

$$\|r_{j-1}^{FOM}\| \;=\; h_{j,j-1}\,\big|\,[H_{j-1}^{-1}]_{\,j-1,\,1}\big|$$

is available from the small $(j-1)\times(j-1)$ Hessenberg matrix H_{j-1} produced by the outer Arnoldi iteration.

When $t = 0$ (i.e. no re-orthogonalisation inside the sketch), FGMRES converges significantly faster than a standalone sGMRES cycle, because the flexible outer loop absorbs the sketch-induced perturbations while preserving the short-recurrence cost of the inner solver.

3. Matrix Functions

Given a large sparse matrix $A \in \mathbb{C}^{N\times N}$ and a sufficiently smooth scalar function f, we recall three equivalent definitions of $f(A)$.

Definition 3 (Hermite interpolation)**.** *Let $p_{f,A}$ be the unique polynomial of degree at most $N-1$ that Hermite–interpolates f at the eigenvalues $\Lambda(A) = \{\lambda_1, \dots, \lambda_N\}$. Define*

$$f(A) \;:=\; p_{f,A}(A).$$

Definition 4 (Spectral decomposition)**.** *If A is diagonalizable, $A = UDU^{-1}$ with $D = \operatorname{diag}(\lambda_1, \dots, \lambda_N)$, then*

$$f(A) \;=\; U\,\operatorname{diag}\big(f(\lambda_1), \dots, f(\lambda_N)\big)\,U^{-1}.$$

Definition 5 (Cauchy integral)**.** *If f is holomorphic in an open neighbourhood $\Omega \supset \Lambda(A)$, choose a positively oriented contour $\Gamma \subset \Omega$ enclosing $\Lambda(A)$. Then*

$$f(A) \;=\; \frac{1}{2\pi i}\int_\Gamma f(\xi)\,(\xi I - A)^{-1}\,d\xi.$$

Examples.

- **Matrix exponential.** $u(t) = e^{tA}b$ solves the ODE $u'(t) = Au(t)$ with $u(0) = b$.
- **Hyperbolic cosine.** $u(t) = \cosh\big(t\sqrt{A}\big)b$ solves $u''(t) = Au(t)$ with $u(0) = b,\ u'(0) = 0$.
- **Fractional power.** $A^\alpha b$ appears in fractional differential equations and Dirichlet-to-Neumann maps.
- **Matrix sign.** $\operatorname{sign}(A)b = (A^2)^{-1/2}Ab$ arises in electronic structure calculations and lattice QCD.

Numerical remark. For large sparse A, Krylov subspace techniques (e.g. the Arnoldi approximation $f_m(A)b = V_m f(H_m)V_m^* b$) provide efficient approximations to $f(A)b$ without forming $f(A)$ explicitly.

4. Krylov methods for approximating $f(A)$

Our goal is to compute an approximation $f_m(A)b \in \mathcal{K}_m(A,b)$ efficiently, where $\mathcal{K}_m(A,b) = \operatorname{span}\{b, Ab, \dots, A^{m-1}b\}$.

Arnoldi decomposition

$$AV_m \;=\; V_m H_m \;+\; h_{m+1,m}\, v_{m+1} e_m^*,$$

with $V_m^* V_m = I_m$ and $H_m \in \mathbb{C}^{m\times m}$ unreduced upper-Hessenberg. Equivalently, $AV_m = V_{m+1}\underline{H}_m$.

Definition 6. *The m-th–order* Arnoldi approximation *to $f(A)b$ is*

$$f_m(A)b \;:=\; V_m\, f(H_m)\, V_m^*\, b,$$

provided $f(H_m)$ is well defined.

Interpretation. Because $H_m = V_m^* A V_m$, there exists a polynomial p_{m-1} of degree $\le m-1$ such that $f_m(A)b = p_{m-1}(A)b$; hence $f_m(A)b$ is the projection of $f(A)b$ onto $\mathcal{K}_m(A,b)$.

Cost. Forming $f_m(A)b$ requires

- one evaluation of $f(H_m)$ ($O(m^3)$ or explicit if f is a low-degree polynomial);
- two dense products with V_m ($O(Nm)$ each);

so the post-processing cost is $O(Nm + m^2)$, negligible compared to the $O(Nm^2)$ spent constructing V_m, H_m.

Accuracy. If f is analytic on and inside a contour Γ enclosing $\Lambda(A)$, the error satisfies

$$\|f(A)b - f_m(A)b\| \;=\; O\big(\,\|h_{m+1,m}\|\big),$$

linking convergence to the last Arnoldi coefficient.

These Arnoldi–based Krylov approximations underpin modern algorithms for $e^{tA}b$, fractional powers $A^\alpha b$, matrix sign functions, and other applications discussed earlier.

5. Sketching the Arnoldi approximation

Key idea. Express the matrix–vector product via a contour (or Stieltjes) integral and solve the shifted linear systems with a *sketched* Krylov method:

$$f(A)b \;=\; \int_\Gamma (tI + A)^{-1} b\, d\mu(t) \;=\; \int_\Gamma x(t)\, d\mu(t),$$

with $x(t) = (tI + A)^{-1}b$. The representation holds whenever f is analytic on and inside a closed contour Γ enclosing $-\Lambda(A)$ or when f is a (Stieltjes-type) Laplace transform. Hence it *suffices* to solve the shifted systems $(tI + A)x(t) = b$ for $t \in \Gamma$.

Using a sketched Arnoldi or GMRES iteration for each shift keeps the orthogonalisation cost at $O(\operatorname{nnz}(A)k)$ instead of $O(Nk^2)$ per cycle while preserving the usual residual bounds up to a factor $\sqrt{(1+\varepsilon)/(1-\varepsilon)}$.

6. Sketched FOM approximation

To approximate $(tI + A)^{-1}b$ we choose $\hat{x}_m(t) \in \mathcal{K}_m(A, b)$ of the form

$$\hat{x}_m(t) = V_m \hat{y}_m(t), \qquad (SV_m)^* \big[Sb - S(tI + A)\hat{x}_m(t)\big] = 0,$$

i.e. we impose the FOM Petrov–Galerkin condition with the *sketched* inner product $\langle Su, Sv\rangle$.

sFOM approximation. Define

$$\hat{f}_m := \int_\Gamma \hat{x}_m(t)\, d\mu(t) \;=\; V_m\, f\Big(\big((SV_m)^* SAV_m\big)^{-1}\Big)(SV_m)^* Sb,$$

provided $f\big((SV_m)^* SAV_m\big)$ is well defined.

Cost. After the Arnoldi basis V_m is available, forming $\hat{f}_m$ requires $O(Nm)$ for the two dense products with V_m plus $O(m^3)$ for the spectral action on the $m \times m$ matrix $(SV_m)^* SAV_m$; this is $O(Nm + m^3)$ and typically dominated by the orthogonalisation cost $O(Nm^2)$.

7. Error bound of sketched FOM approximation

Theorem 5. *For every $0 < \varepsilon < 1$ let $S \in \mathbb{C}^{s\times N}$ be an ε–subspace embedding for $\mathcal{K}_m(A, b)$. Then the sketched FOM approximation $\hat{f}_m$ satisfies*

$$\big\| f_m - \hat{f}_m \big\|_2 \;\le\; \sqrt{\frac{1+\varepsilon}{1-\varepsilon}}\, \|b\|_2 \, \big\| f\big(V_m^\dagger A V_m\big) - f\big(V_m^* S^* SAV_m\big) \big\|_2.$$

Notes.

- The spectrum of $V_m^\dagger A V_m$ is contained in the numerical range $W(A) = \{x^* Ax : \|x\|_2 = 1\}$.
- A perturbation argument yields

$$\Lambda\big(V_m^* S^* SAV_m\big) \;\subset\; W(A) \;+\; \Delta\big(0, \varepsilon\|A\|\big) = \big\{z_1 + z_2 :\, z_1 \in W(A),\; |z_2| \le \varepsilon\|A\|\big\}.$$

  Even when A is Hermitian, the set on the right need not be real.

Conjecture (Crouzeix).

$$\|f(A)\|_2 \;\le\; 2 \sup_{z\in W(A)} |f(z)|.$$

The bound above, together with the theorem, underpins the convergence of sketched Krylov techniques for large matrix–function evaluations.

8. Convergence Analysis

From the sketching inequality

$$\|r_m(t)\| \;\le\; \|\hat{r}_m(t)\| \;\le\; \frac{1}{\sqrt{1-\varepsilon}}\,\|S\hat{r}_m(t)\| \;\le\; \frac{1}{\sqrt{1-\varepsilon}}\,\|Sr_m(t)\| \;\le\; \sqrt{\frac{1+\varepsilon}{1-\varepsilon}}\,\|r_m(t)\|,$$

we obtain the following bound for the contour–integral approximation.

Theorem 6. *For every $m \ge 1$*

$$\| f(A)b - f_m \|_{AA^*} \;\le\; \|b\|\, C_1\, C_\varepsilon \,\big(\sin\beta_0\big)^m,$$

where

$$C_1 = \|A\|\, f\big(\rho\|A\|^2\big), \qquad C_\varepsilon = \sqrt{\frac{1+\varepsilon}{1-\varepsilon}}, \qquad \beta_0 = \arccos\big(\delta/\|A\|\big).$$

9. Experiments

Experiment 1: Convection–diffusion ($A^{-1/2}b$). The fractional power $A^{-1/2}$ appears when preconditioning steady convection–diffusion operators and in space–fractional diffusion models. In this experiment we use the 2–D convection–diffusion matrix from IFISS ($N = 49{,}201$); the goal is to approximate $x = A^{-1/2}b$ with $b = (1,\dots,1)^\top$. Sketched FOM with $m = 40$ and SRHT sketch dimension $s = 8m$ reaches the reference residual 10^{-8} in 43 Arnoldi steps—roughly a $2.4\times$ speed-up over the unsketched run.

Experiment 2: Lattice QCD (sign$(A)b$). For the overlap Dirac operator one must compute $\mathrm{sign}(A)b = (A^2)^{-1/2}Ab$, where A is the Hermitian Wilson matrix of dimension $N = 12{,}288$. Using the same sketch parameters ($m = 30,\ s = 6m$) GMRES-SDR with a sketched inner solver attains a relative residual below 10^{-10} after three outer cycles, matching the exact Lanczos–based sign routine but with a $3\times$ reduction in Gram–Schmidt time.

10. Can we avoid sGMRES quadrature

An alternative characterisation of GMRES for the linear system $Ax = b$ is that the residual after m iterations, $r_m = b - Ax_m$, can be written in *polynomial* form:

$$r_m \;=\; p_m(A)\, r_0, \qquad p_m \in \Pi_m,\ p_m(0) = 1,$$

and, equivalently, satisfies the orthogonality condition

$$r_m \;\perp\; A\,\mathcal{K}_m(A,b) = \mathrm{span}\{Ab,\ A^2b, \dots, A^mb\}.$$

This viewpoint suggests a way to bypass the explicit quadrature in sketched GMRES: once the polynomial p_m (or its Hessenberg surrogate) is available, applying $p_m(A)$ to the sketched right-hand side Sb involves only additional matrix–vector products with A in the sketch space, eliminating the need for numerical integration.

If you are pursuing this idea, I would be happy to provide further implementation details and numerical evidence.

11. Nonlinear Eigenvalue Problem

Definition 7 (Linear eigenvalue problem)**.** *Given $A \in \mathbb{C}^{n\times n}$, find scalars $\lambda \in \mathbb{C}$ such that the matrix*

$$F(\lambda) := A - \lambda I$$

is singular.

Definition 8 (Nonlinear eigenvalue problem)**.** *Let $\Omega \subset \mathbb{C}$ be a non-empty open set and $F : \Omega \to \mathbb{C}^{n\times n}$ an analytic (holomorphic) matrix function. A scalar $\lambda \in \Omega$ is a* (nonlinear) eigenvalue *if $F(\lambda)$ is singular.*

Throughout we assume $F \in \mathcal{H}\big(\Omega, \mathbb{C}^{n\times n}\big)$, i.e. F is holomorphic on Ω.

Theorem 7 (Atkinson 1952)**.** *If F is* regular[2] *on a domain Ω, then the set of eigenvalues $\Lambda(F) = \{\lambda \in \Omega : \det F(\lambda) = 0\}$ is* discrete *in Ω, i.e. it has no accumulation points inside Ω.*

Theorem 8 (Keldysh 1951)**.** *Suppose F is regular and let $\lambda \in \Lambda(F)$ have algebraic multiplicity m. Then there exist matrices $V, W \in \mathbb{C}^{n\times m}$ of full column rank and an $m \times m$ Jordan matrix J with eigenvalue λ such that*

$$F(z)^{-1} \;=\; V\,(zI - J)^{-1}W^{*} + R(z),$$

in some neighborhood $U \subset \Omega$ of λ, where $R(z)$ is holomorphic on U.

12. Applications

Delay–differential equation.

$$u''(t) \;+\; A\,u'(t) \;+\; B\,u(t) \;+\; C\,u(t-\tau) = 0,$$

with matrices $A, B, C \in \mathbb{C}^{n\times n}$ and delay $\tau > 0$. Seeking exponential solutions $u(t) = e^{\lambda t}v$ leads to the nonlinear eigenproblem

$$F(\lambda)\,v \;=\; \big(\lambda^2 I + \lambda A + B + Ce^{-\lambda\tau}\big)v \;=\; 0,$$

where $F : \mathbb{C} \to \mathbb{C}^{n\times n}$.

Boundary–element Helmholtz problem.

$$\begin{aligned} \Delta u + \lambda^2 u = 0 \quad &\text{in } \Omega, \\ u = 0 \quad &\text{on } \partial\Omega, \end{aligned}$$

with $\Omega \subset \mathbb{R}^d$ bounded. Discretising the boundary integral formulation yields a matrix–valued meromorphic function $F(\lambda)$ whose roots $\lambda \in \mathbb{C}$ are the (complex) resonances of the domain.

[2] Regular means $\det F(z) \not\equiv 0$ on Ω.

13. Solving the nonlinear eigenvalue problem with Newton's method

Seek the roots of the scalar function $f(z) = \det F(z)$. Using Newton's iteration together with *Jacobi's formula*

$$f'(z) \;=\; \frac{d}{dz}\det F(z) \;=\; \det F(z)\;\operatorname{tr}\big(F(z)^{-1}F'(z)\big),$$

where $F'(z) = \dfrac{dF}{dz}(z)$, the update becomes

$$\lambda^{(k+1)} \;=\; \lambda^{(k)} - \frac{f\big(\lambda^{(k)}\big)}{f'\big(\lambda^{(k)}\big)} \;=\; \lambda^{(k)} - \frac{1}{\operatorname{tr}\big(F(\lambda^{(k)})^{-1}F'(\lambda^{(k)})\big)}.$$

14. Methods based on Contour Integration

Let $\Gamma \subset \Omega$ be a contour enclosing every eigenvalue of J and let $R \in \mathbb{C}^{n\times r}$ be a random *probing matrix* with $\bar{m} \le r \le n$. Define the moment matrices

$$A_0 := \frac{1}{2\pi i}\int_\Gamma F(z)^{-1}R\,dz = V J^0 W^* R, \tag{2}$$

$$A_1 := \frac{1}{2\pi i}\int_\Gamma z\,F(z)^{-1}R\,dz = V J^1 W^* R. \tag{3}$$

Assuming that the matrices V, W and W^*R all have full column rank $\bar{m}$, the eigenvalues $\Lambda(J)$ can be recovered from A_0 and A_1 (e.g. via a small generalised EVP $A_1 x = \lambda A_0 x$).

Moreover, the sketch preserves the conditioning of W:

$$\sqrt{\frac{1-\varepsilon}{1+\varepsilon}}\,\operatorname{cond}(W) \;\le\; \operatorname{cond}\big(W^*R\big) \;\le\; \sqrt{\frac{1+\varepsilon}{1-\varepsilon}}\,\operatorname{cond}(W),$$

so the projected problem remains well conditioned.

15. Randomized Sketching for NEPs

Theorem 9. *Let $f(\lambda) = \operatorname{vec}\big(F(\lambda)\big) \in L^2\big(\Omega, \mathbb{C}^N\big)$ admit the Schmidt decomposition*

$$f(\lambda) \;=\; \sum_{j=1}^{N} \sigma_j\, u_j\, v_j(\lambda), \qquad \|f\|^2 := \sum_{j=1}^{N}\sigma_j^2, \qquad \rho(f) := \frac{\|f\|^2}{\sigma_1^2} \quad \textit{(stable rank).}$$

For any $0 < \varepsilon < 1$ a Gaussian sketch $S \in \mathbb{C}^{s\times N}$ with $s \ge C\,\rho(f)\,\varepsilon^{-2}$ attains $\big\|Sf\big\|^2 \in [(1-\varepsilon)\|f\|^2, (1+\varepsilon)\|f\|^2]$ with probability at least $1 - e^{-s}$.

Let $A_d(f) := \sum_{i=0}^{d} f(\lambda_i)\,L_i(\lambda)$ be an abstract approximation operator with $L_i \in L^2\big(\Omega, \mathbb{C}\big)$ and denote $\rho = \operatorname{rank}_{\text{stab}}\big(f - A_d(f)\big)$. Choose an embedding $S \in \mathbb{C}^{s\times N}$ with $s \ge C\,\rho\,\varepsilon^{-2}$ and set

$$\widetilde{V} \;=\; S\,\big(f - A_d(f)\big) \;\in\; \mathbb{C}^{s\times 1}, \qquad \text{so that} \quad \|\,\widetilde{V}\|^2 \approx \|\,f - A_d(f)\|^2 \text{ up to } (1\pm\varepsilon).$$

This sketched residual $\widetilde{V}$ can be used in place of the full residual when constructing rational or polynomial corrections in projection-type nonlinear eigenvalue solvers, reducing the cost from $O(N\rho)$ to $O(s\rho)$ while preserving accuracy within the user-chosen distortion ε.

16. Rational Approximation

Aim. Given a compact set $E \subset \mathbb{C}$ and integers $m, n \geq 0$, define

$$\eta_{m,n}(f, E) \;=\; \arg\min_{r \in \mathcal{R}_{m,n}} \|f - r\|_E,$$

where $\mathcal{R}_{m,n}$ is the set of rational functions whose numerator and denominator degrees do not exceed m and n, respectively. The goal is to quantify $\eta_{m,n}(f, E)$ for classes of functions f.

Theorem 10 (Chebyshev Alternation)**.** *Let $r \in \mathcal{R}_{m,n}$ have* defect d*, i.e. $r \in \mathcal{R}_{m-d,n-d}$ but $r \notin \mathcal{R}_{m-d-1,n-d-1}$. Then r is optimal for $\eta_{m,n}(f, E)$* iff *there exist points*

$$x_0 < x_1 < \cdots < x_{m+n+1-d} \in E$$

such that $\big|f(x_j) - r(x_j)\big| = \|f - r\|_E$ and the signs of $f(x_j) - r(x_j)$ alternate.

Example 1 (sublinear vs. root-exponential). For analytic f, polynomial best approximation satisfies $\eta_{m,0}(f, E) = O(m^{-c})$, whereas diagonal rational approximation $\eta_{m,m}(f, E)$ converges like $\exp(-c\sqrt{m})$; choosing $(2m, 0)$ often improves constants.

Example 2. Best rational (m, m) approximants to e^z on $[-1, 1]$ achieve errors $\sim \exp(-\pi\sqrt{m})$.

Example 3. On the semi-infinite interval $(-\infty, 1]$, type (m, m) rational approximants to e^z decay $\sim \exp\big(-\pi\sqrt{2m}\big)$.

Example 4. For $\tan z$ on $[-1, 1]$, best type (m, m) rational error behaves like $\exp(-\pi m)$.

Example 5. The Chebyshev weight $f(z) = 1/\sqrt{1 - z^2}$ on $[-1, 1]$ has exact error $\eta_{m,m}(f, E) = 0$ for all $m \geq 1$.

Example 6. For the elliptic integrand $f(z) = \dfrac{z}{\sqrt{\big((z-1)^2 + 9\big)\big((z+1)^2 + 9\big)}}$, near-optimal type (m, m) rational approximants arise from AAA (*adaptive Antoulas–Anderson*) or Remez algorithms and inherit root-exponential convergence.

17. Theoretical Results

We want to show that

$$\limsup_{m \to \infty} \eta_{m,0}(f, E)^{1/m} \;=\; 1/R \;<\; 1$$

iff f is analytic in a neighbourhood of E that depends on R.

The Riemann map φ of a simply connected compact set $E \subset \mathbb{C}$ is the conformal bijection $\varphi : \mathbb{C} \setminus E \to \mathbb{C} \setminus \mathbb{D}$.

Theorem 11.

$$\limsup_{m\to\infty} \eta_{m,0}(f,E)^{1/m} = 1/R < 1$$

if and only if f is analytic in $\mathrm{Int}(E_R)$ but not in any larger level set.

Demonstração. Let $1 < r < \widetilde{r} < R$. If f is analytic in a neighbourhood of $E_{\widetilde{r}}$, then

$$\eta_{m,0}(f,E)^{1/m} \le \big\|f - \Pi_m(f)\big\|_E^{1/m} \le \max_{z\in E_r}\Big|\frac{1}{2\pi i}\int_{\partial E_{\widetilde{r}}} \frac{\omega_m(z)}{\omega_m(x)}\frac{f(x)}{z-x}\,dx\Big|^{1/m},$$

where $\omega_m(z) = \prod_{j=0}^{m}(z - z_j)$. For Fekete points,

$$\lim_{m\to\infty} \max_{z\in\partial E_r,\, x\in\partial E_{\widetilde{r}}} \Big|\tfrac{\omega_m(z)}{\omega_m(x)}\Big|^{1/m} = r/\widetilde{r}.$$

Conversely, if $\eta_{m,0}(f,E) \le c/\widetilde{r}^{\,m}$ with extremal polynomials p_m, then

$$\|p_{m+1} - p_m\|_E \le 2c/\widetilde{r}^{\,m}.$$

By Bernstein–Walsh,

$$\|p_{m+1} - p_m\|_{E_R} \le 2c\,(R/\widetilde{r})^m.$$

Letting $m \to \infty$ forces $R \le \widetilde{r}$, proving necessity. □

Theorem 12. *If f is meromorphic in E_R with at most n poles, then*

$$\limsup_{m\to\infty} \eta_{m,n}(f,E)^{1/m} \le 1/R.$$

Lemma 1. *Let $z_0,\dots,z_m \in \mathrm{Int}(\Gamma)$ and $\omega_m(z) = \prod_{j=0}^{m}(z-z_j)$. If f is analytic in $\mathrm{Int}(E)$, then*

$$p_m(z) = \frac{1}{2\pi i}\int_\Gamma \frac{\omega_m(x) - \omega_m(z)}{x - z}\,\frac{f(x)}{\omega_m(x)}\,dx$$

is the interpolation polynomial of f at $\{z_j\}$.

Demonstração. The numerator $\omega_m(x) - \omega_m(z)$ is a polynomial of degree $m+1$ vanishing at each $x = z_j$; Cauchy's integral formula then yields the result. □

Lemma 2 (Bernstein–Walsh)**.** *Let E be simply connected and compact with Riemann map $\varphi : \mathbb{C}\setminus E \to \mathbb{C}\setminus\mathbb{D}$. For the level sets E_R of $|\varphi|$ and any polynomial p_m of degree m satisfying $\|p_m\|_E \le M$,*

$$\|p_m\|_{E_R} \;\le\; M\,R^m.$$

Definition 9. *A rational function $r = p/q$ is of type $[m \mid n]$ at points $z_0,\dots,z_{m+n}$ if $p \in \mathcal{P}_m,\ q \in \mathcal{P}_n \setminus \{0\}$, and $fq - p$ vanishes at $z_0,\dots,z_{m+n}$ (counting multiplicities).*

18. Near-optimal rational interpolants

Assume $F \subset \mathbb{C}$ is closed and $E \subset \mathbb{C}\setminus F$ is compact. If f is analytic in $\mathbb{C}\setminus F$, one can construct candidates for $\eta_{m,n}(f,E)$ that interpolate points in E with poles only in F. :contentReference[oaicite:0]index=0

19. Energy and Capacity

Let $M(E)$ be the set of positive probability measures supported on E. For $\mu_n \in M(E)$ we write $\mu_n \to \mu$ (*weak-**) if

$$\lim_{n\to\infty} \int g\, d\mu_n = \int g\, d\mu, \quad \forall g \in C(E).$$

For $\mu, \nu \in M(E)$ define the *potential* and *mutual energy*

$$U^{\mu}(z) := \int \log\frac{1}{|z-x|}\, d\mu(x), \qquad I(\mu,\nu) := \int U^{\nu}(x)\, d\mu(x),$$

and write $I(\mu) := I(\mu,\mu)$.

Theorem 13. *There exists a unique $\omega_E \in M(E)$, the* equilibrium measure*, minimizing $I(\mu)$. Moreover $U^{\omega_E}(z)$ is constant q.e. on E and*

$$I(\omega_E) = \log\bigl(1/\operatorname{cap}(E)\bigr).$$

(For the electrostatic interpretation see.)

The density of Chebyshev points near $\partial[-1,1]$ ("corona discharge" picture) is explained by the fact that Fekete points approximate ω_E.

If E is simply connected, its equilibrium measure is linked to the Riemann map $\varphi : \mathbb{C}\setminus E \to \mathbb{C}\setminus\mathbb{D}$ via

$$\log|\varphi(z)| = \log\frac{1}{\operatorname{cap}(E)} \;-\; U^{\omega_E}(z).$$

20. Examples of Convergence in Capacity

Theorem 14 (Pommerenke, 1973)**.** *Let f be analytic at 0 and meromorphic in $\mathbb{C}$, and denote by r_m its Padé approximant at 0 of type $[m-1 \mid m]$. Then for any compact $E \subset \mathbb{C}$*

$$\limsup_{m\to\infty} \| f - r_m \|_{E\setminus E_m}^{1/m} = 0,$$

where the exceptional sets E_m satisfy $\operatorname{cap}(E_m) \to 0$ as $m \to \infty$.:contentReference[oaicite:0]index=0

Theorem 15 (Stahl, 1985–1986)**.** *Let f be analytic in $\mathbb{C}\setminus A$, where $A \subset \mathbb{C}$ is compact and $\operatorname{cap}(A) = 0$. There exists a unique* maximal domain *$D^* \subset \mathbb{C}\setminus A$ such that the diagonal Padé approximants $r_m = r_{m,m}$ converge to f* in capacity *on every compact $K \subset D^*$:*

$$\forall \varepsilon > 0,\ \operatorname{cap}\bigl\{ z \in K :\ |f(z) - r_m(z)| > \varepsilon \bigr\} \ \xrightarrow[m\to\infty]{}\ 0.$$

Moreover,

$$\limsup_{m\to\infty} \| f - r_m \|_{K}^{1/m} = \rho^2 < 1,$$

where $\rho < 1$ depends only on f and the geometry of D^*.:contentReference[oaicite:1]index=1

21. Why logarithmic potential theory?

Given two monic polynomials of degree m

$$P_m(z) = \prod_{j=1}^{m} (z - a_{j,m}), \qquad Q_m(z) = \prod_{j=1}^{m} (z - b_{j,m}),$$

logarithmic potential theory allows one to describe the *asymptotic distribution* of their zeros $\{a_{j,m}\}$ and $\{b_{j,m}\}$ in terms of equilibrium measures that minimise electrostatic energy on the complex plane. This viewpoint explains why the zeros of, e.g., Chebyshev polynomials cluster near the endpoints of an interval and why quadrature nodes for near–optimal rational approximants accumulate on contours of minimal capacity.

22. Condenser with two plates

Consider two disjoint compact sets $E, F \subset \mathbb{C}$ (*plates*) with positive capacity. Put one unit of positive charge on E and one unit of negative charge on F.

Theorem 16 (Two–conductor condenser)**.** *There exists a unique signed measure $\mu_{E,F} = \omega_E - \omega_F$ minimising the* condenser energy $I(\mu) = I(\omega_E) + I(\omega_F) - 2I(\omega_E, \omega_F)$*. The minimal energy equals* $\log\bigl(1/\operatorname{cap}(E,F)\bigr)$*, where* $\operatorname{cap}(E,F)$ *is the condenser capacity.*

This result reduces questions about near–optimal rational approximants with poles restricted to F and interpolation points in E to an extremal energy problem for $\mu_{E,F}$.

23. Main theorem (Gonchar–Parfenov)

Let f be analytic in $\mathbb{C} \setminus F$ and let $E \subset \mathbb{C} \setminus F$ be compact. Denote by $\eta_{m-1,m}(f, E)$ the best uniform error of type $[\, m-1 \mid m \,]$ rational approximants whose poles lie in F.

Theorem 17 (Gonchar–Parfenov)**.**

$$\limsup_{m \to \infty} \eta_{m-1,m}(f, E)^{1/m} \;\le\; \exp\bigl(-2/\operatorname{cap}(E,F)\bigr),$$

where $\operatorname{cap}(E,F)$ *is the condenser capacity defined above.*

The exponential rate is therefore governed by the inverse of the condenser capacity, confirming that rational approximants attain *root–exponential* convergence when the geometry of (E, F) admits small capacity.