

# Flow Models and Flow Matching and Transformations and Ansatzes in Quantum Many-Body Physics

Graduate-level report (XeLaTeX)

December 30, 2025

**Abstract.** This report has two goals. **Part I** reviews the fundamentals of *flow-based* generative modeling: the change-of-variables theorem, common architectural building blocks (coupling layers, autoregressive flows, invertible convolutions, and spline layers), and the continuous-time limit (continuous normalizing flows). We then explain *flow matching* and *conditional flow matching*, which train continuous-time flows without differentiating through ODE solvers. We also discuss modeling *discrete* data with flows, including dequantization, integer/discrete coupling, and surjective constructions, with an emphasis on pros/cons and practical trade-offs.

**Part II** surveys a wide set of *transformations* and *variational ansatzes* used across quantum many-body physics, especially in condensed matter and quantum information. We provide a table that organizes these methods by goal (constraints, correlations, mappings, diagonalization, decoupling, dualities, and modern variational families), and then derive each method in detail: Gutzwiller projection, Jastrow correlators, Laughlin and composite-fermion constructions, Jordan–Wigner and Bravyi–Kitaev mappings, Bogoliubov transformations, match-gates/fermionic Gaussian states, Hubbard–Stratonovich decouplings, Schrieffer–Wolff canonical transformations, Kramers–Wannier duality, bosonization, spin-boson mappings (Holstein–Primakoff and Schwinger bosons), tensor-network ansatzes, coupled-cluster methods, and neural-network quantum states.

---

# Contents

---

---

## List of Figures

---

---

## List of Tables

---

## Part I

# Flow Models and Flow Matching

# Preliminaries: Change of Variables and Flow Objectives

## 1.1 Change of variables and exact likelihoods

A *normalizing flow* constructs a complicated probability density by pushing forward a simple base density through an *invertible* mapping. Let  $\mathbf{z} \in \mathbb{R}^d$  be a latent variable with known density  $p_Z(\mathbf{z})$  (e.g. standard Gaussian), and let  $f_\theta : \mathbb{R}^d \rightarrow \mathbb{R}^d$  be a diffeomorphism (invertible, differentiable, differentiable inverse). The data variable is

$$\mathbf{x} = f_\theta(\mathbf{z}), \quad \mathbf{z} = f_\theta^{-1}(\mathbf{x}). \quad (1.1)$$

Then the induced density on  $\mathbf{x}$  is given by the change-of-variables theorem:

$$p_X(\mathbf{x}) = p_Z(f_\theta^{-1}(\mathbf{x})) \left| \det \frac{\partial f_\theta^{-1}(\mathbf{x})}{\partial \mathbf{x}} \right|. \quad (1.2)$$

Equivalently, in log form,

$$\log p_X(\mathbf{x}) = \log p_Z(\mathbf{z}) - \log \left| \det \frac{\partial f_\theta(\mathbf{z})}{\partial \mathbf{z}} \right| \Big|_{\mathbf{z}=f_\theta^{-1}(\mathbf{x})}. \quad (1.3)$$

A dataset  $\{\mathbf{x}^{(n)}\}_{n=1}^N$  can thus be trained by exact maximum likelihood (minimizing negative log-likelihood). The two computational bottlenecks in (??) are:

1. efficient evaluation of  $f_\theta^{-1}(\mathbf{x})$  (the inverse map), and
2. efficient evaluation of the log-determinant of the Jacobian.

Flow architectures are therefore designed to make these operations tractable.

## 1.2 A taxonomy of flow layers

Flow layers can be organized by how they achieve tractability:

- **Coupling layers** (NICE/RealNVP/Glow): exact inverse and triangular Jacobian; expressive via stacking and permutation.
- **Autoregressive flows** (MAF/IAF): triangular Jacobian but sequential structure; either sampling or density is fast, not both.

- **Invertible linear transforms:** permutations, invertible  $1 \times 1$  convolutions, orthogonal/PLU parameterizations.
- **Elementwise monotone transforms:** splines, rational-quadratic couplings.
- **Continuous-time flows (CNFs):** define  $\dot{\mathbf{x}} = v_\theta(t, \mathbf{x})$ ; log-density changes by divergence.

?? summarizes common layer types.

Layer family	Key idea	Cost characteristics
Coupling (additive/affine)	Split variables $\mathbf{x} = (\mathbf{x}_a, \mathbf{x}_b)$ ; leave $a$ unchanged; transform $b$ using a NN conditioned on $a$ . Jacobian is triangular.	Fast inverse and log-det; requires multiple layers for mixing.
Autoregressive (MAF/IAF)	Each dimension transformed conditioned on previous ones. Jacobian triangular by construction.	One direction is sequential (slow).
Invertible linear (permute, $1 \times 1$ conv)	Mix channels/dimensions between nonlinear layers to improve expressivity.	Log-det often cheap (e.g. $\log  \det W $ ).
Spline/monotone elementwise	Replace affine maps by monotone splines (e.g. rational-quadratic).	More expressive, still tractable inverse and log-det.
Residual / ODE (CNF)	Define continuous dynamics $\dot{\mathbf{x}} = v(t, \mathbf{x})$ ; log-density evolves via divergence.	Training/sampling require ODE solves unless using flow matching.

Table 1.1: A pragmatic taxonomy of flow layers and their computational trade-offs.

### 1.3 Visual intuition: a coupling layer

?? illustrates the architecture of an affine coupling layer.

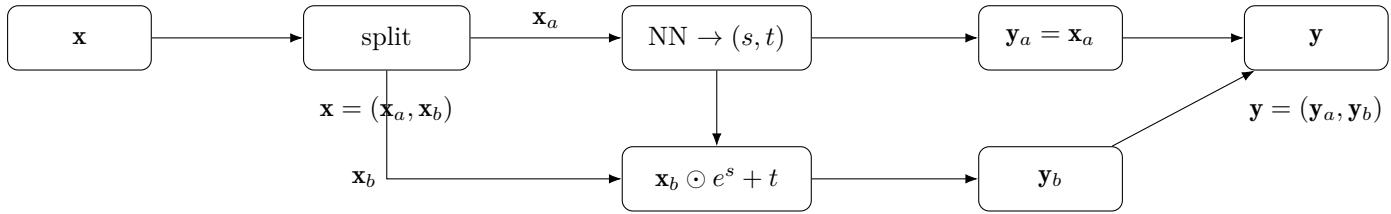


Figure 1.1: Affine coupling layer:  $\mathbf{y}_a = \mathbf{x}_a$ ,  $\mathbf{y}_b = \mathbf{x}_b \odot \exp s(\mathbf{x}_a) + t(\mathbf{x}_a)$ . Inverse is immediate:  $\mathbf{x}_a = \mathbf{y}_a$ ,  $\mathbf{x}_b = (\mathbf{y}_b - t(\mathbf{y}_a)) \odot \exp[-s(\mathbf{y}_a)]$ .

# Discrete-time Flows: RealNVP, Glow, Autoregressive and Spline Flows

## 2.1 Coupling layers: NICE and RealNVP

Coupling layers originate from NICE [? ] and RealNVP [? ]. The simplest form is *additive coupling*:

$$\mathbf{y}_a = \mathbf{x}_a, \quad \mathbf{y}_b = \mathbf{x}_b + t_\theta(\mathbf{x}_a). \quad (2.1)$$

The Jacobian is triangular with unit diagonal, hence *volume preserving* ( $\log\det = 0$ ). RealNVP generalizes to an *affine* coupling

$$\mathbf{y}_a = \mathbf{x}_a, \quad \mathbf{y}_b = \mathbf{x}_b \odot \exp(s_\theta(\mathbf{x}_a)) + t_\theta(\mathbf{x}_a), \quad (2.2)$$

so that

$$\log \left| \det \frac{\partial \mathbf{y}}{\partial \mathbf{x}} \right| = \sum_i s_{\theta,i}(\mathbf{x}_a). \quad (2.3)$$

Why is this expressive? A single coupling layer only changes half the variables, so we stack many such layers and alternate the split (or apply permutations) so that every dimension is eventually modified. In image models, the split is often a channel-wise split, and the NN is a convolutional network.

## 2.2 Multi-scale architectures and squeezing

Flow-based image models commonly use a *multi-scale* architecture: after some transformations, part of the variables are “factored out” as latent variables, reducing the dimensionality of subsequent layers. This improves efficiency and encourages the model to represent coarse features early and fine details later. In RealNVP, a “squeeze” operation reshapes spatial dimensions into channels (e.g.  $H \times W \times C \mapsto (H/2) \times (W/2) \times 4C$ ), enabling channel-wise coupling to capture local pixel correlations.

## 2.3 Glow and invertible $1 \times 1$ convolutions

Glow [? ] improves RealNVP by introducing:

1. **ActNorm**: an affine per-channel normalization initialized from data (like batch norm but deterministic at test time), and
2. **Invertible  $1 \times 1$  convolutions**: a learned linear mixing of channels per spatial location, parameterized by an invertible matrix  $W \in \mathbb{R}^{C \times C}$ .

For an invertible  $1 \times 1$  convolution applied to each spatial location, the log-determinant is  $HW \log |\det W|$ . Glow parameterizes  $W$  via an LU decomposition so that  $\log |\det W|$  is cheap to compute.

## 2.4 Autoregressive flows: MAF and IAF

Autoregressive flows use the factorization  $p(\mathbf{x}) = \prod_i p(x_i | x_{<i})$  but in an invertible transform. A typical autoregressive transform is

$$y_i = \mu_i(\mathbf{x}_{<i}) + \sigma_i(\mathbf{x}_{<i}) x_i, \quad (2.4)$$

which yields a triangular Jacobian. Masked Autoregressive Flow (MAF) [? ] makes density evaluation fast (because all  $y_i$  can be computed in parallel), but inversion/sampling is sequential. Inverse Autoregressive Flow (IAF) [? ] swaps the roles to make sampling fast but density slow. This illustrates a general trade-off: autoregressive structure enforces triangular Jacobians but introduces sequential dependence in at least one direction.

## 2.5 Spline flows and monotone transforms

Affine couplings can be replaced by more expressive monotone elementwise transforms. Neural spline flows [? ] use rational-quadratic splines whose parameters are predicted by a NN. Because the spline is monotone on each interval, it is invertible; both the inverse and the log-derivative can be computed analytically. In practice, spline couplings improve likelihood and sample quality at similar computational cost.

## 2.6 Residual flows and invertible ResNets

Another design is to use residual connections while enforcing invertibility. A residual flow has the form

$$\mathbf{y} = \mathbf{x} + g_\theta(\mathbf{x}), \quad (2.5)$$

which is invertible if  $g_\theta$  is a contraction (e.g. Lipschitz constant  $< 1$ ). Then the Jacobian determinant can be approximated via power-series of  $\log \det(I + J_g)$  and trace estimators. Residual flows connect to continuous-time flows and ODE interpretations.

# Continuous Normalizing Flows and Flow Matching

## 3.1 Continuous-time limit and the log-density ODE

A continuous normalizing flow (CNF) defines dynamics

$$\frac{dx(t)}{dt} = v_\theta(t, \mathbf{x}(t)), \quad t \in [0, 1], \quad (3.1)$$

with  $\mathbf{x}(0) \sim p_0$  and  $\mathbf{x}(1)$  as the generated sample. The instantaneous change of log-density along trajectories is

$$\frac{d}{dt} \log p_t(\mathbf{x}(t)) = -\nabla \cdot v_\theta(t, \mathbf{x}(t)). \quad (3.2)$$

Derivation sketch: write the continuity equation  $\partial_t p_t + \nabla \cdot (p_t v) = 0$ , then take the material derivative along a trajectory and use  $\frac{d}{dt} p_t(\mathbf{x}(t)) = \partial_t p_t + v \cdot \nabla p_t$ .

Likelihood evaluation thus requires integrating (??) and (??). CNFs such as FFJORD [?] use Hutchinson trace estimators to approximate  $\nabla \cdot v$  without forming the full Jacobian, reducing cost from  $O(d^2)$  to roughly  $O(d)$  per evaluation.

## 3.2 Why flow matching?

Training CNFs by maximum likelihood can be expensive because it involves repeated ODE solves and backpropagation through the solver (often via the adjoint method). **Flow Matching** (FM) [?] reframes training as *supervised vector-field regression* that does not require solving the generative ODE during training.

The starting point is a *probability path*  $\{p_t\}_{t \in [0,1]}$  connecting  $p_0$  (noise) to  $p_1$  (data). Any sufficiently smooth path admits a velocity field  $u^*(t, \mathbf{x})$  satisfying the continuity equation

$$\partial_t p_t(\mathbf{x}) + \nabla \cdot (p_t(\mathbf{x}) u^*(t, \mathbf{x})) = 0. \quad (3.3)$$

If we learn  $u_\theta \approx u^*$ , then integrating  $\dot{\mathbf{x}} = u_\theta(t, \mathbf{x})$  transports  $p_0$  approximately into  $p_1$ .

## 3.3 Flow matching objective

FM chooses a tractable way to sample from  $p_t$  and to compute (or sample) the target velocity. A particularly useful instantiation is **conditional flow matching** (CFM): sample an endpoint

$\mathbf{x}_1 \sim q_1$  from the data distribution, sample  $\mathbf{x}_0 \sim p_0$  from the base distribution, and define a conditional interpolation  $\mathbf{x}_t = \phi_t(\mathbf{x}_0 | \mathbf{x}_1)$ . A simple choice is linear interpolation

$$\mathbf{x}_t = (1 - t)\mathbf{x}_0 + t\mathbf{x}_1, \quad \dot{\mathbf{x}}_t = \mathbf{x}_1 - \mathbf{x}_0. \quad (3.4)$$

Then we can train  $u_\theta$  via mean-square regression

$$\mathcal{L}_{\text{FM}}(\theta) = \mathbb{E}_{t \sim \text{Unif}[0,1]} \mathbb{E}_{\mathbf{x}_0 \sim p_0, \mathbf{x}_1 \sim q_1} \|u_\theta(t, \mathbf{x}_t) - (\mathbf{x}_1 - \mathbf{x}_0)\|^2. \quad (3.5)$$

More sophisticated paths (e.g. diffusion-type paths or optimal-transport paths) can reduce stiffness and improve sample quality. Lipman *et al.* show that FM can match or exceed diffusion models on image generation tasks while enabling ODE-based sampling and likelihood evaluation [? ? ].

### 3.4 Relation to diffusion models and score matching

Diffusion models define a forward noising process that maps data  $\mathbf{x}_1 \sim q$  to noise  $\mathbf{x}_0 \sim p_0$  by an SDE, and train a score network to reverse it [? ? ?]. Flow matching can be viewed as learning a deterministic vector field whose trajectories approximate a chosen probability path. When the path is chosen as the marginal of an SDE, the FM objective becomes closely related to denoising-score matching, but with a regression target that is often more numerically stable.

### 3.5 A minimal algorithmic template

1. Sample  $\mathbf{x}_1 \sim q_1$  (data),  $\mathbf{x}_0 \sim p_0$  (noise), and  $t \sim \text{Unif}[0, 1]$ .
2. Compute  $\mathbf{x}_t = \phi_t(\mathbf{x}_0 | \mathbf{x}_1)$  (e.g. ??).
3. Regress  $u_\theta(t, \mathbf{x}_t)$  to the known target velocity (e.g.  $\mathbf{x}_1 - \mathbf{x}_0$ ).
4. After training, sample by solving  $\dot{\mathbf{x}} = u_\theta(t, \mathbf{x})$  from  $t = 0$  to 1 with initial  $\mathbf{x}(0) \sim p_0$ .

Flow matching thus replaces expensive “simulate-then-differentiate” loops with “sample-and-regress” training.

# Discrete Data and Discrete Flows

## 4.1 Why discrete data is subtle

The change-of-variables formula (??) relies on a Lebesgue density and a Jacobian determinant. For purely discrete variables  $x \in \{0, 1\}^N$  or  $x \in \{0, \dots, 255\}^N$ , probability is defined with respect to counting measure, so a deterministic bijection  $f$  merely permutes probability mass. If the base distribution is factorized or uniform, a bijection alone cannot create arbitrary target probabilities.

This observation motivates three broad strategies:

1. **Dequantization:** embed discrete data in continuous space by adding noise.
2. **Stochastic / surjective flows:** allow non-invertible maps augmented with auxiliary noise.
3. **Truly discrete flows:** define invertible transformations in a discrete algebra (e.g. modular arithmetic) and combine with expressive base distributions.

## 4.2 Dequantization for images

For integer pixel data  $\mathbf{x} \in \{0, \dots, 255\}^D$ , define a continuous variable

$$\mathbf{y} = \mathbf{x} + \mathbf{u}, \quad \mathbf{u} \sim \text{Unif}([0, 1]^D). \quad (4.1)$$

Train a continuous density model  $p_\theta(\mathbf{y})$ ; this yields a lower bound on the discrete likelihood of  $\mathbf{x}$  by integrating  $p_\theta$  over the unit hypercube around  $\mathbf{x}$ . Flow++ [? ] introduced *variational dequantization*: instead of uniform noise, learn a conditional noise distribution  $q_\phi(\mathbf{u} | \mathbf{x})$  using an auxiliary flow, tightening the bound and improving likelihood.

## 4.3 Discrete flows: invertible transforms on finite alphabets

Discrete normalizing flows aim to model  $p(\mathbf{x})$  exactly without dequantization. A common design is a discrete coupling layer for  $K$ -ary variables  $x \in \{0, \dots, K - 1\}^D$  using modular arithmetic:

$$\mathbf{y}_a = \mathbf{x}_a, \quad \mathbf{y}_b = (\mathbf{x}_b + t_\theta(\mathbf{x}_a)) \bmod K. \quad (4.2)$$

This is invertible because subtraction mod  $K$  is well-defined. Such constructions appear in integer discrete flows and discrete flows for sequences [? ? ]. The main challenge is expressiveness:

because there is no Jacobian term, the model must rely on an expressive *base* distribution and deep compositions.

## 4.4 Binary data: flows on $\{0, 1\}^N$

For binary vectors, modular addition is XOR. A binary coupling can be written as

$$\mathbf{y}_a = \mathbf{x}_a, \quad \mathbf{y}_b = \mathbf{x}_b \oplus t_\theta(\mathbf{x}_a), \quad (4.3)$$

where  $t_\theta(\mathbf{x}_a) \in \{0, 1\}^{|b|}$ . This is invertible since XOR is its own inverse. However, XOR couplings alone are limited because they cannot change Hamming weight patterns smoothly; expressive modeling often requires adding latent noise (surjective flows) or hybridizing with autoregressive components.

## 4.5 Surjective flows and latent-variable constructions

Surjective/augmented flows relax bijectivity by introducing auxiliary variables  $\mathbf{u}$  so that the map  $(\mathbf{x}, \mathbf{u}) \mapsto \mathbf{z}$  is bijective, but  $\mathbf{x} \mapsto \mathbf{z}$  is not. This allows true changes in probability mass. SurVAE flows [? ] unify VAEs and flows by combining surjective and stochastic layers with tractable likelihood bounds. In practice, surjective flows can model discrete data more flexibly than purely bijective discrete flows.

## 4.6 Pros and cons summary

Approach	Pros	Cons
Dequantization	Leverages mature continuous-flow toolkit; scalable to images; good likelihoods.	Not an exact discrete model; choice of dequantization distribution affects bound.
Pure discrete bijective flows	Exact likelihood on discrete space; no Jacobian computations.	Expressiveness limited unless base distribution is rich; design/inference can be hard at large alphabet size.
Surjective / stochastic flows	Can change mass and model complex discreteness; unifies flows and VAEs.	Usually gives bounds not exact likelihood; additional variance and design complexity.

Table 4.1: High-level trade-offs in modeling discrete data with flow-based methods.

## Flows in Physics: Symplectic and Equivariant Constraints (Brief)

---

Flows have also been adapted to respect physical structure. In Hamiltonian mechanics, canonical transformations preserve the symplectic form; Li *et al.* proposed *symplectic flows* as neural canonical transformations [? ]. In molecular modeling, equivariant flows enforce  $E(3)$  symmetry (rotations/translations) to model configurations. These ideas echo a recurring theme in physics: powerful transformations are those that preserve the relevant geometry (symplectic, gauge, or symmetry constraints).

## **Part II**

# **Transformations and Ansatzes in Quantum Many-Body Physics**

# Overview and Summary Table

## 6.1 Why transformations and ans”atze matter

Quantum many-body Hilbert spaces grow exponentially. Two broad strategies help:

1. **Transformations:** rewrite the Hamiltonian or degrees of freedom into a form where the physics is clearer or the problem is solvable (exactly or approximately). Examples include mappings between spins and fermions (Jordan–Wigner), auxiliary-field decouplings (Hubbard–Stratonovich), and canonical transformations (Schrieffer–Wolff).
2. **Variational ans”atze:** propose a family of wavefunctions with manageable complexity that captures key correlations (Gutzwiller, Jastrow, Laughlin) or obeys entanglement constraints (tensor networks).

These tools are ubiquitous: from integrable 1D models to quantum Hall physics, from superconductivity to quantum simulation on qubits.

## 6.2 A roadmap table

Table 6.1: Survey table of transformations and ans”atze covered in Part II. The “Type” column labels the dominant role: mapping/transform (T) or variational ansatz (A).

Method	Type	Core idea and typical use
Gutzwiller projector	A	Enforce strong local constraints (e.g. no double occupancy) by projecting a mean-field wavefunction. Useful for Hubbard/t-t-J and spin liquids.
Jastrow correlator / Feenberg	A	Multiply a reference state by $\exp(\sum_{i < j} u(r_{ij}))$ to incorporate pair correlations and cusp conditions.
Laughlin wavefunction	A	Holomorphic Jastrow polynomial in LLL producing incompressible FQH liquids; leads to fractional charge/anyons.
Composite fermions / T/A flux attachment	A	Singular gauge transformation attaching $2p$ flux quanta (Chern–Simons); maps FQH fractions to integer QH of composite fermions.

Method	Type	Core idea and typical use
Jordan–Wigner mapping	T	Map 1D spins to fermions via a nonlocal parity string; solves Ising/XY chains and encodes fermions on qubits.
Bravyi–Kitaev mapping	T	Alternative fermion-to-qubit transform with shorter Pauli strings (logarithmic locality) for quantum simulation.
Bogoliubov (BdG) transformation	T	Canonical linear transform mixing $c$ and $c^\dagger$ to diagonalize quadratic Hamiltonians (BCS, superfluids).
Matchgates and fermionic Gaussian states	T	Quadratic fermionic unitaries correspond to orthogonal transforms of Majoranas; matchgate circuits are classically simulable.
Hubbard–Stratonovich decoupling	T	Rewrite quartic interactions as integrals over auxiliary fields; basis for mean-field and determinantal QMC.
Schrieffer–Wolff canonical transform	T	Unitary perturbative elimination of high-energy subspace; yields effective low-energy Hamiltonians (t-J, Kondo).
Particle–hole transformation	T	Exchange particles and holes; reveals symmetries (e.g. bipartite Hubbard) and simplifies pairing problems.
Kramers–Wannier duality	T	Map Ising order operators to disorder operators; exposes self-dual criticality and dual phases.
Bosonization	T	Map 1D fermions/spins to bosonic fields; yields Luttinger liquids, universal exponents.
Holstein–Primakoff spin-wave expansion	T/A	Represent spins as bosons about an ordered state; derive magnon dispersions and $1/S$ corrections.
Schwinger bosons / partons	T/A	Express spins in terms of fractionalized bosons/fermions with gauge constraints; build spin-liquid ansatzes.
Tensor networks (MPS/PEPS/MERA)	A	Variational families structured by entanglement; MPS for 1D gapped systems, PEPS for 2D, MERA for criticality.
Coupled cluster and unitary coupled cluster	A	Exponential ansatz $e^T \Phi\rangle$ (or $e^{T-T^\dagger}$ ) for correlated ground states; central in quantum chemistry and many-body theory.
Neural-network quantum states	A	Parameterize amplitudes by neural nets (RBM, autoregressive, equivariant); flexible variational ansatz for correlated phases.

# Gutzwiller Projection and Jastrow Correlators

## 7.1 Gutzwiller projector: definition and physical meaning

Consider the Hubbard model

$$H = -t \sum_{\langle ij \rangle, \sigma} (c_{i\sigma}^\dagger c_{j\sigma} + \text{h.c.}) + U \sum_i n_{i\uparrow} n_{i\downarrow}. \quad (7.1)$$

At large  $U/t$ , double occupancy is energetically costly. A variational way to encode this is the **Gutzwiller wavefunction** [?]:

$$|\Psi_G\rangle = P_G(g) |\Psi_0\rangle, \quad P_G(g) = \prod_i [1 - (1-g) n_{i\uparrow} n_{i\downarrow}], \quad (7.2)$$

where  $|\Psi_0\rangle$  is typically a Slater determinant or BCS state and  $g \in [0, 1]$  tunes the weight of doubly occupied sites. The limit  $g = 0$  enforces the constraint  $n_{i\uparrow} n_{i\downarrow} = 0$  on every site, appropriate for the  $U \rightarrow \infty$  Mott regime.

### 7.1.1 Projecting a Slater determinant: from metal to Mott

If  $|\Psi_0\rangle$  is a Fermi sea, then  $|\Psi_G\rangle$  is a highly correlated state. The projection entangles the degrees of freedom because it correlates occupations on each site. One can see this by writing the amplitude of a configuration  $|\{n_{i\uparrow}, n_{i\downarrow}\}\rangle$ :

$$\Psi_G(\{n\}) = g^{D(\{n\})} \Psi_0(\{n\}), \quad D(\{n\}) = \sum_i n_{i\uparrow} n_{i\downarrow}. \quad (7.3)$$

Thus the projector is a simple scalar weight  $g^D$  in the occupation basis.

### 7.1.2 Gutzwiller approximation: renormalization factors

A commonly used analytical approximation replaces expectation values in  $|\Psi_G\rangle$  by renormalizing hopping and exchange processes. For example, a hopping process  $c_{i\sigma}^\dagger c_{j\sigma}$  is suppressed because it can create double occupancy. In the Gutzwiller approximation, one estimates

$$\langle c_{i\sigma}^\dagger c_{j\sigma} \rangle_G \approx g_t \langle c_{i\sigma}^\dagger c_{j\sigma} \rangle_0, \quad (7.4)$$

with  $g_t$  a doping-dependent factor. For the  $t$ - $J$  model at hole doping  $\delta$  (electron density  $1 - \delta$ ) one obtains

$$g_t = \frac{2\delta}{1 + \delta}, \quad g_J = \frac{4}{(1 + \delta)^2}, \quad (7.5)$$

where  $g_J$  renormalizes superexchange terms  $\mathbf{S}_i \cdot \mathbf{S}_j$ . These formulas can be derived by simple counting of allowed local configurations under the projection (probability that a hop is allowed in the projected ensemble divided by that in the unprojected ensemble). While uncontrolled, the approximation is qualitatively useful and underlies renormalized mean-field theories of cuprates.

## 7.2 Jastrow correlators: from cusp conditions to long-range order

A **Jastrow** wavefunction multiplies a reference state by an explicit correlator [? ? ]:

$$|\Psi_J\rangle = \exp\left(\frac{1}{2} \sum_{i \neq j} u_{ij} \hat{O}_i \hat{O}_j\right) |\Phi\rangle. \quad (7.6)$$

In continuum electron systems,  $\hat{O}_i$  is often the density at position  $\mathbf{r}_i$  and  $u_{ij} = u(|\mathbf{r}_i - \mathbf{r}_j|)$ . For lattice models, common choices are density-density Jastrow factors  $\exp(\sum_{i < j} v_{ij} n_i n_j)$  or spin Jastrow factors.

### 7.2.1 Deriving the electron-electron cusp condition

For Coulomb-interacting electrons, the wavefunction must satisfy a cusp condition as two electrons approach. Consider two electrons with relative coordinate  $r$  and reduced mass  $\mu = m/2$ . Near  $r = 0$ , the Schrödinger equation in relative coordinates is dominated by

$$\left(-\frac{1}{2\mu} \frac{d^2}{dr^2} + \frac{e^2}{r}\right) \psi(r) \approx E\psi(r). \quad (7.7)$$

Neglecting  $E$  and higher terms, one finds that  $\psi'(0^+)/\psi(0) = \mu e^2$  (up to spin-dependent factors). A Jastrow factor with  $u(r) \sim ar$  at small  $r$  can enforce this cusp by making  $\psi(r) \sim e^{-u(r)/2} \approx 1 - (a/2)r + \dots$ .

### 7.2.2 Long-range Jastrow and Luttinger liquids

In 1D, long-range Jastrow correlators can generate Luttinger-liquid exponents. For example, a wavefunction with

$$u_{ij} \propto -K \log \left| \sin \frac{\pi(i-j)}{L} \right| \quad (7.8)$$

produces power-law correlations characteristic of a Luttinger parameter  $K$ . This foreshadows the deep connections between Jastrow forms, bosonization, and conformal field theory wavefunctions.

# Quantum Hall Ansatz: Laughlin and Composite Fermions

## 8.1 Lowest Landau level and holomorphic structure

For a charged particle in a perpendicular magnetic field, the kinetic energy forms Landau levels separated by  $\hbar\omega_c$ . In the lowest Landau level (LLL), wavefunctions in symmetric gauge take the form

$$\psi(z, \bar{z}) = f(z) \exp\left(-\frac{|z|^2}{4\ell_B^2}\right), \quad z = x + iy, \quad (8.1)$$

where  $f(z)$  is holomorphic. Many-body fermionic states in the LLL are therefore antisymmetric holomorphic polynomials times the Gaussian.

## 8.2 Laughlin wavefunction and filling fraction

The Laughlin ansatz at filling  $\nu = 1/m$  (odd  $m$  for fermions) is [? ]:

$$\Psi_m(z_1, \dots, z_N) = \prod_{i < j} (z_i - z_j)^m \exp\left(-\sum_k \frac{|z_k|^2}{4\ell_B^2}\right). \quad (8.2)$$

The factor  $(z_i - z_j)^m$  ensures an  $m$ -th order zero when particles coincide, suppressing close encounters and lowering Coulomb energy. On the sphere, the relation between number of flux quanta  $N_\phi$  and particle number is  $N_\phi = m(N - 1)$ ; thus  $\nu \approx N/N_\phi \rightarrow 1/m$  for large  $N$ .

### 8.2.1 Quasiholes and fractional charge

A quasihole at position  $\eta$  is created by

$$\Psi_{\text{qh}}(\eta) = \prod_i (z_i - \eta) \Psi_m. \quad (8.3)$$

By adiabatically inserting one flux quantum, one produces a deficit of charge  $e/m$ . The Berry phase for braiding two quasiholes yields an anyonic statistical phase  $\theta = \pi/m$ .

### 8.2.2 Plasma analogy (sketch)

$|\Psi_m|^2$  can be interpreted as the Boltzmann weight of a 2D one-component plasma at inverse temperature  $\beta = 2m$ . This analogy explains screening and incompressibility, and provides intuition for fractionalization.

## 8.3 Flux attachment and composite fermions

A powerful transformation for FQH physics is **flux attachment**: attach  $2p$  flux quanta to each electron via a singular gauge transformation, mapping interacting electrons at filling  $\nu$  to *composite fermions* at an effective filling. In first-quantized form, define

$$\Psi_{\text{CF}} = \prod_{i < j} \left( \frac{z_i - z_j}{|z_i - z_j|} \right)^{2p} \Psi_e, \quad (8.4)$$

which multiplies the wavefunction by a phase that depends on particle positions and changes the effective statistics of the attached flux.

Field-theoretically, this corresponds to introducing a Chern–Simons gauge field  $a_\mu$  with constraint

$$\frac{1}{2\pi} \epsilon^{ij} \partial_i a_j = 2p \rho, \quad (8.5)$$

so that the CS magnetic field is proportional to density. In mean-field,  $\rho$  is replaced by its average so composite fermions see an effective magnetic field

$$B^* = B - 2p \rho \phi_0, \quad (8.6)$$

with  $\phi_0 = h/e$ . If composite fermions fill  $n$  effective Landau levels, one obtains Jain fractions

$$\nu = \frac{n}{2pn \pm 1}. \quad (8.7)$$

This transformation provides a unifying picture of many observed FQH states [? ? ].

# Spin–Fermion and Fermion–Qubit Mappings

## 9.1 Jordan–Wigner transformation in 1D

On a 1D chain, define fermion operators by

$$c_j = \left( \prod_{k < j} \sigma_k^z \right) \sigma_j^-, \quad c_j^\dagger = \left( \prod_{k < j} \sigma_k^z \right) \sigma_j^+. \quad (9.1)$$

The nonlocal string ensures fermionic anticommutation: for  $i < j$ ,  $\{c_i, c_j\} = 0$  because the string in  $c_j$  contains  $\sigma_i^z$  which anticommutes with  $\sigma_i^-$ . One also finds  $\sigma_j^z = 1 - 2c_j^\dagger c_j$ .

### 9.1.1 Example: transverse-field Ising to Kitaev chain

The 1D transverse-field Ising model

$$H = -J \sum_j \sigma_j^x \sigma_{j+1}^x - h \sum_j \sigma_j^z \quad (9.2)$$

maps under Jordan–Wigner to a quadratic fermion Hamiltonian with pairing terms. In Majorana variables  $\gamma_{j,1} = c_j + c_j^\dagger$ ,  $\gamma_{j,2} = i(c_j^\dagger - c_j)$ , one obtains the Kitaev chain form, revealing Majorana edge modes in the ordered phase.

## 9.2 Bravyi–Kitaev transformation

Jordan–Wigner strings have length  $O(N)$ , which is costly for quantum simulation. The Bravyi–Kitaev (BK) transformation [? ?] balances locality of parity and occupancy information, reducing typical Pauli-string lengths to  $O(\log N)$ . BK can be understood as encoding each fermionic mode’s occupation and parity in a binary tree structure. The explicit mapping can be described via three sets (update, parity, flip sets) defining which qubits store which parity bits. While more complex, BK often reduces circuit depth for chemistry Hamiltonians.

# Quadratic Fermions: Bogoliubov and Matchgates

## 10.1 General quadratic Hamiltonians and Nambu space

A general fermionic quadratic Hamiltonian can be written as

$$H = \sum_{ij} c_i^\dagger A_{ij} c_j + \frac{1}{2} \sum_{ij} (c_i^\dagger B_{ij} c_j^\dagger + \text{h.c.}), \quad (10.1)$$

where  $A$  is Hermitian and  $B$  is antisymmetric. Define the Nambu spinor  $\Psi = (c_1, \dots, c_N, c_1^\dagger, \dots, c_N^\dagger)^T$ . Then

$$H = \frac{1}{2} \Psi^\dagger \begin{pmatrix} A & B \\ -B^* & -A^T \end{pmatrix} \Psi + \text{const.} \quad (10.2)$$

Diagonalization amounts to finding a linear transformation  $\Psi = W\Gamma$  that preserves canonical anticommutation. Such  $W$  are elements of a Bogoliubov group (a subgroup of  $U(2N)$  with particle-hole constraints).

## 10.2 Bogoliubov transformation and BCS

For each momentum pair  $(k, -k)$  in BCS mean-field, the Hamiltonian reduces to a  $2 \times 2$  Bogoliubov–de Gennes block. The canonical transform

$$\gamma_k = u_k c_k + v_k c_{-k}^\dagger, \quad (10.3)$$

with  $|u_k|^2 + |v_k|^2 = 1$ , diagonalizes the block, yielding quasiparticle energies  $E_k = \sqrt{\xi_k^2 + |\Delta|^2}$  [? ? ].

## 10.3 Majorana representation and Gaussian states

Define Majorana operators

$$\gamma_{2j-1} = c_j + c_j^\dagger, \quad \gamma_{2j} = i(c_j^\dagger - c_j), \quad (10.4)$$

which satisfy  $\{\gamma_a, \gamma_b\} = 2\delta_{ab}$ . Quadratic Hamiltonians generate orthogonal transformations on Majoranas: a Gaussian unitary  $U = \exp(-\frac{i}{4}\gamma^T K \gamma)$  acts as

$$U^\dagger \gamma U = R\gamma, \quad R \in SO(2N). \quad (10.5)$$

Gaussian states are fully characterized by the covariance matrix  $\Gamma_{ab} = \frac{i}{2}\langle[\gamma_a, \gamma_b]\rangle$ .

## 10.4 Matchgates as fermionic Gaussian circuits

A **matchgate** is a two-qubit gate that preserves fermionic Gaussianity under the Jordan–Wigner mapping. Valiant showed that nearest-neighbor matchgate circuits are classically simulable [?]; Jozsa and Miyake clarified the connection to fermionic linear optics [? ]. The simulation proceeds by updating the covariance matrix under each local orthogonal transformation.

# Auxiliary-Field Transformations: Hubbard–Stratonovich

## 11.1 Gaussian integral identity

The Hubbard–Stratonovich (HS) transformation uses the Gaussian identity

$$\exp\left(\frac{\lambda}{2}O^2\right) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} d\phi \exp\left(-\frac{\phi^2}{2} + \sqrt{\lambda}\phi O\right), \quad \lambda > 0. \quad (11.1)$$

This linearizes a quadratic operator  $O^2$  at the cost of integrating over an auxiliary field  $\phi$ .

## 11.2 Decoupling the Hubbard interaction

In determinantal QMC, one Trotter-decomposes  $e^{-\beta H}$  into time slices and decouples the on-site interaction  $Un_{\uparrow}n_{\downarrow}$  via an HS field. A standard discrete HS (Hirsch) writes

$$e^{-\Delta\tau U(n_{\uparrow}-\frac{1}{2})(n_{\downarrow}-\frac{1}{2})} = \frac{1}{2} \sum_{s=\pm 1} e^{\alpha s(n_{\uparrow}-n_{\downarrow})}, \quad \cosh \alpha = e^{\Delta\tau U/2}. \quad (11.2)$$

Then fermions become quadratic for fixed  $\{s_{i,\tau}\}$ , allowing integration and yielding a determinant weight [? ? ].

## 11.3 Mean-field as saddle point

The same HS transformation underlies mean-field theory: in a path integral, one introduces a field  $\phi(\tau)$  and approximates the integral by its saddle point  $\phi \approx \phi_0$ . The saddle-point condition yields the mean-field self-consistency equation (e.g. BCS gap equation when decoupling in the pairing channel).

# Canonical Transformations: Schrieffer–Wolff and Particle–Hole

## 12.1 Schrieffer–Wolff: general idea

Suppose  $H = H_0 + V$ , where  $H_0$  has a large gap separating low-energy (L) and high-energy (H) subspaces, and  $V$  weakly couples them. The Schrieffer–Wolff (SW) transformation seeks a unitary  $e^S$  (with  $S$  anti-Hermitian) such that

$$H_{\text{eff}} = e^S H e^{-S} \quad (12.1)$$

has no matrix elements between L and H to a desired order. Expanding,

$$H_{\text{eff}} = H + [S, H] + \frac{1}{2}[S, [S, H]] + \dots \quad (12.2)$$

Choosing  $S$  so that  $V + [S, H_0] = 0$  (off-diagonal part) eliminates coupling at first order, producing an effective Hamiltonian in L at second order.

## 12.2 Application: Hubbard to t–J

At large  $U$ , take  $H_0 = U \sum_i n_{i\uparrow} n_{i\downarrow}$  and  $V$  the hopping. Eliminating virtual double-occupancy processes yields the t–J model

$$H_{tJ} = -t \sum_{\langle ij \rangle, \sigma} \tilde{c}_{i\sigma}^\dagger \tilde{c}_{j\sigma} + J \sum_{\langle ij \rangle} \left( \mathbf{s}_i \cdot \mathbf{s}_j - \frac{1}{4} n_i n_j \right), \quad J = \frac{4t^2}{U}, \quad (12.3)$$

where  $\tilde{c}$  act in the no-double-occupancy subspace. The exchange term arises from a second-order process where an electron hops to form a doubly occupied intermediate state then hops back, producing an effective antiferromagnetic coupling.

## 12.3 Particle–hole transformation on bipartite lattices

On a bipartite lattice, the Hubbard model has a particle–hole symmetry at half filling. Define

$$c_{i\downarrow} \mapsto (-1)^i c_{i\downarrow}^\dagger, \quad (12.4)$$

where  $(-1)^i$  is  $+1$  on one sublattice and  $-1$  on the other. This maps repulsive  $U$  to attractive  $U$  and relates antiferromagnetism to  $\eta$ -pairing superconductivity. Such transformations are invaluable for proving sign-problem free conditions and mapping phases.

# Dualities and Field Redefinitions: Kramers–Wannier and Bosonization

## 13.1 Kramers–Wannier duality of the 1D quantum Ising chain

Consider the quantum Ising chain

$$H = -J \sum_j \sigma_j^z \sigma_{j+1}^z - h \sum_j \sigma_j^x. \quad (13.1)$$

Define dual spins  $\mu_{j+1/2}^x = \sigma_j^z \sigma_{j+1}^z$  and  $\mu_{j+1/2}^z = \prod_{k \leq j} \sigma_k^x$ . One checks that  $\mu$  satisfy Pauli algebra. In terms of  $\mu$ ,

$$H = -J \sum_j \mu_{j+1/2}^x - h \sum_j \mu_{j-1/2}^z \mu_{j+1/2}^z. \quad (13.2)$$

Thus the model is self-dual under  $J \leftrightarrow h$ . The critical point is at  $J = h$  in the thermodynamic limit, matching the exact solution (and reflecting the original Kramers–Wannier duality [?]).

## 13.2 Bosonization in 1D: from fermions to a compact boson

In 1D, low-energy fermions near the Fermi points can be decomposed into right- and left-movers,  $\psi(x) = e^{ik_F x} \psi_R(x) + e^{-ik_F x} \psi_L(x)$ . Define chiral boson fields  $\phi_{R/L}$  via the density modes  $\rho_{R/L} = \pm \frac{1}{2\pi} \partial_x \phi_{R/L}$ . One obtains the bosonization identity

$$\psi_r(x) \sim \frac{1}{\sqrt{2\pi a}} \eta_r \exp(irk_F x + i[r\phi(x) - \theta(x)]), \quad r = \pm, \quad (13.3)$$

where  $\phi$  and  $\theta$  are bosonic fields,  $a$  is a short-distance cutoff, and  $\eta_r$  are Klein factors. The interacting fermion Hamiltonian maps to the Luttinger liquid form

$$H = \frac{v}{2\pi} \int dx \left[ K(\partial_x \theta)^2 + \frac{1}{K} (\partial_x \phi)^2 \right], \quad (13.4)$$

with  $K$  encoding interactions. Correlation functions become power laws with exponents set by  $K$  [? ?].

# Spin–Boson Representations: Holstein–Primakoff and Schwinger Bosons

## 14.1 Holstein–Primakoff transformation and spin waves

For a spin- $S$  operator,

$$S^z = S - a^\dagger a, \quad S^+ = \sqrt{2S - a^\dagger a} a, \quad S^- = a^\dagger \sqrt{2S - a^\dagger a}. \quad (14.1)$$

Expanding the square root in  $1/S$  yields spin-wave theory. For a ferromagnetic Heisenberg model  $H = -J \sum_{\langle ij \rangle} \mathbf{S}_i \cdot \mathbf{S}_j$ , the quadratic Hamiltonian after HP and Fourier transform gives magnon dispersion  $\omega_{\mathbf{k}} = 2JS(z - \gamma_{\mathbf{k}})$ .

## 14.2 Antiferromagnets: two sublattices and Bogoliubov diagonalization

For bipartite antiferromagnets, one rotates spins on one sublattice, applies HP on each, and keeps quadratic terms. The resulting bosonic Hamiltonian contains terms like  $a_{\mathbf{k}} b_{-\mathbf{k}} + \text{h.c.}$  and is diagonalized by a bosonic Bogoliubov transform, producing gapless Goldstone modes.

## 14.3 Schwinger bosons and parton constraint

Schwinger bosons represent a spin as two bosons  $b_{i\uparrow}, b_{i\downarrow}$ :

$$\mathbf{S}_i = \frac{1}{2} b_i^\dagger \boldsymbol{\sigma} b_i, \quad \sum_{\sigma} b_{i\sigma}^\dagger b_{i\sigma} = 2S. \quad (14.2)$$

The constraint introduces a local gauge redundancy. Mean-field decouplings yield spin-liquid ansatzes with emergent gauge fields, and projection back to the physical Hilbert space yields variational states.

# Modern Variational Families: Tensor Networks, Coupled Cluster, Neural States

## 15.1 Matrix product states from Schmidt decomposition

For a 1D chain, repeatedly applying the Schmidt decomposition across each bond yields a matrix product state (MPS) representation

$$|\Psi\rangle = \sum_{\{s_i\}} \text{tr}(A_{s_1}^{[1]} A_{s_2}^{[2]} \cdots A_{s_L}^{[L]}) |s_1 \cdots s_L\rangle. \quad (15.1)$$

If the entanglement entropy across each cut is bounded (area law), then a modest bond dimension suffices. DMRG optimizes over MPS efficiently and is the gold standard for 1D gapped systems [? ? ].

## 15.2 PEPS and MERA (brief)

Projected entangled pair states (PEPS) generalize MPS to higher dimensions; multiscale entanglement renormalization ansatz (MERA) captures scale invariance and criticality.

## 15.3 Coupled cluster and unitary coupled cluster

Coupled cluster uses an exponential ansatz  $|\Psi\rangle = e^T |\Phi_0\rangle$  with  $T = T_1 + T_2 + \cdots$  containing particle-hole excitations. Projecting the Schrödinger equation onto excited determinants yields nonlinear equations for cluster amplitudes. Unitary coupled cluster (UCC) uses  $|\Psi\rangle = e^{T-T^\dagger} |\Phi_0\rangle$  and is prominent in quantum algorithms (VQE) because it is generated by a Hermitian operator.

## 15.4 Neural-network quantum states

Neural-network quantum states parameterize amplitudes as  $\Psi_\theta(\mathbf{s})$  using networks. The restricted Boltzmann machine (RBM) ansatz [? ] writes

$$\Psi(\mathbf{s}) = \sum_{\mathbf{h}} \exp(\mathbf{a} \cdot \mathbf{s} + \mathbf{b} \cdot \mathbf{h} + \mathbf{s}^T W \mathbf{h}), \quad (15.2)$$

and marginalizing hidden units yields a correlator-product form closely related to Jastrow-like factors. Autoregressive neural wavefunctions provide exact sampling without MCMC and connect conceptually to flow models.

## Concluding Remarks

---

Flow models and many-body transformations share a philosophy: *choose an invertible (or controlled) change of variables that makes the structure simple*. In machine learning, the goal is tractable likelihoods and sampling; in physics, it is diagonalization, decoupling, or building correlated trial states. Recent cross-pollination—symplectic flows, equivariant generative models, and neural variational ansatzes—suggests that this shared viewpoint will remain fruitful.

## Appendix: Useful Identities

---

### A.1 Hutchinson trace estimator

For a matrix  $A$ ,  $\text{tr}(A) = \mathbb{E}_{\epsilon}[\epsilon^T A \epsilon]$  when  $\epsilon$  has i.i.d. Rademacher entries  $\pm 1$ .