Flow Field Chess: A Deliberately Unfit Engine for a Genuinely Fun Game

Gregor Hubert Max Koch*

Preprint — June 29, 2026

Abstract

This paper presents the Flow Field Chess Engine (FFCE), a chess-playing system constructed entirely from mathematical algorithms that were never designed for chess. The engine resolves scalar fields—pressure, resistance, trace, and flow—computed from piece geometry, then selects moves by greedy descent on an energy functional defined over those fields. It uses no opening book, no endgame tablebases, and no lookahead search: every move is chosen by single-ply evaluation of the resulting field configuration. By conventional metrics it is a weak engine, and deliberately so.

We argue that the dominant paradigm of chess AI—deep search, large opening databases, and statistical optimisation—optimises for strength at the expense of *engagement*. FFCE instead aims for an opponent that is fallible in legible ways and interesting to play against, with no pretension to competitive strength. Our contribution is a novel, deliberately non-competitive field-based chess engine together with its mathematics: we describe the full field-resolution pipeline and give a complete formalism for the five core fields and thirteen "borrowed" algorithms (Gray–Scott reaction–diffusion, Ising spin lattices, Lattice Boltzmann advection, quaternion trace, and others). We illustrate the resulting play with two exhibition games against a large language model, used purely to show *how* the engine moves—its emergent, field-driven choices and legible blunders—rather than how strongly. We make no empirical strength claim; the central design idea (that superposing chess-seeded but chess-agnostic algorithms yields emergent, legible play) is offered as a hypothesis and a working system, not a benchmarked result.

Keywords: chess engine, flow fields, potential fields, emergent behaviour, game AI, reaction–diffusion, Ising model, influence maps, non-adversarial search, human-centred game design.

1 Introduction: The Problem with Perfection

1.1 The omniscient chess engine

By 1997 Deep Blue had defeated Garry Kasparov [2]; by 2017 AlphaZero had defeated Stockfish in self-play matches [18]. In the narrow sense of "which move is best," competitive chess is effectively a solved engineering problem. Modern engines such as Stockfish [20] evaluate positions at superhuman depth, and tablebases provide perfect play for all positions with seven or fewer pieces.

This is a remarkable achievement. It is also, for most recreational players, of limited recreational value. Playing a full-strength engine is less a contest than a demonstration: the human makes moves and the engine makes better ones, with the outcome essentially predetermined.

*Independent researcher, GameDev.Tech. Contact: `gregor.koch@maxstudios.de`.

1.2 The broken psychological contract

The standard remedy is to handicap the engine—reducing search depth, adding noise, or limiting time. This produces *weaker* play but not necessarily *more interesting* play. A throttled search still "knows" the strong move and declines to play it; its blunders carry no internal logic. We contend that pretending to lose is psychologically distinct from genuinely trying and failing, and that recreational players can feel the difference.

1.3 Thesis

We propose that a recreational chess engine should optimise for *authenticity* rather than strength. We take an authentic opponent to be one that (i) sees the same board the human sees, with no hidden databases; (ii) reasons from inspectable principles rather than opaque weights; (iii) makes mistakes that follow from that reasoning rather than from injected noise; and (iv) produces games that feel contested regardless of outcome.

FFCE pursues this by abandoning adversarial search entirely. It evaluates the current position through a collection of physical and mathematical fields, each expressing a different "opinion" about the board, and plays the move that most improves the aggregate field state. This is an unusual design, and the bulk of this paper is devoted to specifying it precisely (Sections 2 and 3), reporting what we have actually measured (Section 4), and discussing what those measurements do and do not support (Section 5).

A note on scope and tone. The genesis of this project is openly playful: it began as the question "what happens if we throw algorithms from chemistry and physics at a chessboard?" [10]. We have retained some of that voice because it reflects the design intent, but we have taken care to phrase the project's larger claims—about omniscience, engagement, and the suitability of large models for recreational play—as arguments and hypotheses rather than as empirically established findings. The reader should treat Sections 5 and the conclusion as position-taking commentary grounded in, but not proven by, the limited experiments reported here.

2 Methodology: Mathematics Masquerading as Chess

2.1 The five core fields

FFCE computes five foundational scalar fields from every position. Informally:

Pressure (P)
: the net control map, White influence minus Black influence; positive values are White territory, negative Black, and the gradients mark contested frontiers.

Attack (A)
: raw projection of piece influence; each piece stamps a movement kernel onto the board and the sum is a reach map.

Resistance (R)
: positional friction; occupancy, enemy control, and king proximity raise the cost of influence flowing through a square.

Trace (T)
: directional memory; recent moves deposit "intent" along their paths via per-piece quaternion headings.

Table 1: The thirteen extra algorithms and their reinterpretation. References are to the originating literature, not to chess applications.

Algorithm	Original domain	Chess interpretation
Reaction–diffusion	Chemical patterns [21, 6]	Emerging pressure hotspots
Cellular automata	Biological growth [5]	Stable vs. unstable control
Resistor network	Electrical engineering	Equilibrium influence flow
Ising spin	Statistical physics [11, 8]	Clustered advantage alignment
Wave resonance	Acoustics	Resonant attack corridors
Lattice Boltzmann	Fluid dynamics [13]	Slow-building pressure currents
Spectral low-freq.	Signal processing	Global positional trends
Hodge curl proxy	Vector calculus [1]	Rotational trap potential
Ant pheromone	Swarm intelligence [3, 4]	Long-term influence memory
Fuzzy future	Kernel density [16, 15]	Gaussian presence uncertainty
Topo. persistence	Topological data analysis	Stable connected territories
Latent channels	ML embeddings	Synthetic feature compression
Tensor kernel	Kernel density [16, 15]	Soft power distribution

Flow (F)
diffused influence; raw attack is iteratively spread into neighbouring squares, damped by resistance.

In isolation each is a reasonable chess heuristic (control, mobility, coordination). The novelty is computing them as literal scalar fields resolved through physical equations rather than hand-tuned evaluation terms. The precise definitions appear in Section 3.

2.2 The thirteen "borrowed" algorithms

We augment the core fields with thirteen additional algorithms drawn from domains unrelated to chess (Table 1). None of these methods "knows" chess: reaction–diffusion forms chemical patterns, the Ising model describes ferromagnetism, Lattice Boltzmann simulates fluids. We seed each with chess-relevant data, run it on the board, and read out the resulting 8×8 field. The design conjecture—examined critically in Section 5—is that such chess-agnostic processes, when seeded and aggregated appropriately, produce chess-relevant structure.

2.3 The field-resolution pipeline

The core contribution of FFCE is not any individual algorithm but the *pipeline* that makes disparate mathematical tools chess-relevant and combines them into a single evaluation.

Step 1: board encoding. The 8×8 board is encoded as parallel arrays: an occupancy matrix $O[x, y] \in \{-6, \ldots, 6\}$ (piece type, sign for colour), per-colour raw attack tensors $A_{\text{white}}, A_{\text{black}}$, and king coordinates K_w, K_b. Each piece projects influence according to its movement rule (Section 3.2).

Step 2: core fields. Pressure, resistance, trace, and flow are computed from the attack tensors as specified in Sections 3.3–3.6.

Step 3: extra-algorithm adaptation. Each extra algorithm receives the board state and core fields and returns an 8×8 scalar field. The key design rule is that *every algorithm is seeded from*

chess-relevant data. For example, the Gray–Scott activator is initialised as $V_0 = 0.5 + 0.1\,P$ (seeded from pressure), so its emergent spots bloom where pressure is high; Ising spins are seeded from $\mathrm{sign}(P)$ with an external field pointing toward the enemy king, so clusters form preferentially in attacking zones. Full specifications are in Section 3.9 and Appendix A.

Step 4: aggregation. All fields are combined into a single score by averaging each field over the 3×3 zone $Z(K_{\text{enemy}})$ around the enemy king and taking a weighted sum,

$$\text{Score} \;=\; \sum_i w_i\, \Phi_i(Z(K_{\text{enemy}})), \tag{1}$$

where the weights w_i are adjustable at run time via the GUI. The king-centric aggregation is what implicitly defines the objective: configurations that project influence toward the enemy king score highly.

Step 5: move selection. For each legal move, FFCE simulates the move, recomputes all fields, and scores the result via (1); it plays the highest-scoring move. This is *single-ply* evaluation—no search tree, no minimax, no alpha–beta. The equivalent energy-minimisation view is given in Section 3.7.

2.4 Why this can produce chess-like behaviour

Three design principles are intended to make non-chess algorithms yield chess-relevant signal.

1. **Chess-relevant seeding.** Every algorithm is initialised from chess-meaningful data (piece positions, attack patterns, movement history), so it operates on chess-relevant inputs even though its dynamics are domain-agnostic.

2. **King-centric aggregation.** The score averages field values near the enemy king (Eq. 1), implicitly defining "good" as "projecting influence toward the enemy king."

3. **Multi-algorithm superposition.** Any single algorithm is idiosyncratic; superposing many is intended to let agreements reinforce while uncorrelated disagreements cancel (the variance argument of Section 3.8).

We state the third principle as a *hypothesis*: that sufficient superposition of chess-seeded algorithms yields emergent tactical awareness. The evidence we can offer for it (Section 4) is suggestive but limited.

2.5 The one-time symmetry break

At the starting position all fields are symmetric and White and Black have identical influence. To break this deadlock without hard-coding an opening, FFCE applies a single random perturbation to the initial quaternion orientations. After move one there is no further randomness: subsequent play is deterministic, though sensitive to initial conditions in the manner of a chaotic system.

3 Mathematical Formalism

This section gives the complete formalism underlying FFCE. A reader content with the informal description of Section 2 may skip to Section 4.

Table 2: Piece weights from the released code (`constants.py`). "Material" is used for capture/value accounting; "Attack" weights scale the influence kernel of Eq. (6).

	Pawn	Knight	Bishop	Rook	Queen	King
Material weight	1.0	3.2	3.3	5.0	9.0	0.0
Attack weight	1.2	3.4	3.4	5.2	9.4	2.0

3.1 The board as a discrete manifold

Let $\mathcal{B} = \{(x, y) : x, y \in \{0, \ldots, 7\}\}$ denote the 64-square board as a discrete 2-dimensional lattice with true (non-periodic) boundaries. The discrete Laplacian is

$$(\Delta f)_{x,y} \;=\; \frac{1}{4} \sum_{(x',y') \in \mathcal{N}(x,y)} f_{x',y'} \;-\; f_{x,y}, \tag{2}$$

where $\mathcal{N}(x, y)$ is the von Neumann (4-connected) neighbourhood, boundary squares having reduced neighbour counts. The gradient is

$$(\nabla f)_{x,y} \;=\; \left(\tfrac{f_{x+1,y} - f_{x-1,y}}{2},\; \tfrac{f_{x,y+1} - f_{x,y-1}}{2} \right), \tag{3}$$

with one-sided differences at boundaries.

3.2 Piece-influence kernels

Each piece type $\tau \in \{P, N, B, R, Q, K\}$ defines an influence kernel $\kappa_\tau : \mathcal{B} \times \mathcal{B} \to \mathbb{R}$. For pawns and knights,

$$\kappa_P\big((x_s, y_s), (x_t, y_t)\big) = \begin{cases} 1 & |x_t - x_s| = 1 \;\wedge\; y_t - y_s = \sigma \\ 0 & \text{otherwise,} \end{cases} \tag{4}$$

$$\kappa_N\big((x_s, y_s), (x_t, y_t)\big) = \begin{cases} 1 & (|x_t - x_s|, |y_t - y_s|) \in \{(1,2), (2,1)\} \\ 0 & \text{otherwise,} \end{cases} \tag{5}$$

with $\sigma = +1$ for White and -1 for Black. Bishops, rooks and queens project along unobstructed rays (diagonal, orthogonal, and their union, respectively, with a per-step decay applied to sliders); the king acts in all eight adjacent directions. The raw attack field for colour c is the superposition

$$A_c(x, y) \;=\; \sum_{p \in \text{pieces}_c} w_{\tau(p)}\; \kappa_{\tau(p)}\big(\text{pos}(p), (x, y)\big). \tag{6}$$

Piece weights. Table 2 lists the material and attack weights actually used in the released implementation (`constants.py`). The illustrative integer weights used in some equations of the source notes ($w_N = 3$, $w_B = 3.2$, $w_R = 5$) are rounded approximations of these; Table 2 is authoritative.

3.3 The pressure field

The pressure field $P : \mathcal{B} \to [-1, 1]$ represents net territorial control,

$$P(x, y) \;=\; \tanh\big(\beta_P \,(A_{\text{white}}(x, y) - A_{\text{black}}(x, y))\big), \qquad \beta_P = 0.7. \tag{7}$$

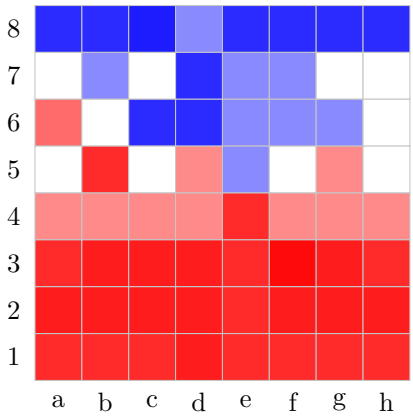

Figure 1: Pressure field $P(x, y) = \tanh(0.7\,(A_w - A_b))$ for the position after `1.e4 e5 2.Nf3 Nc6 3.Bb5 a6 4.Ba4 Nf6` (Ruy Lopez). Red is White control, blue is Black control, white is contested. Values reproduced verbatim from the source notes. Ranks 1–3 form a solid White base; rank 8 is Black territory; ranks 4–7 are contested.

The hyperbolic tangent bounds the field to $[-1, 1]$ (preventing runaway accumulation in heavily contested squares) while preserving maximal sensitivity near zero pressure, i.e. on contested frontiers. Figure 1 shows the resulting field for a Ruy Lopez position, rendered directly from the values tabulated in the source notes.

Proposition 3.1 (Static pressure conservation)**.** *In the absence of piece movement, the total unsigned pressure* $\sum_{(x,y)} |P(x, y)|$ *is constant in time.*

Proof. P is a deterministic function of piece positions alone; with positions fixed, P is fixed. The statement becomes non-trivial only once flow dynamics (Section 3.5) couple the field to a time parameter. □

3.4 The resistance field

The resistance field $R : \mathcal{B} \to \mathbb{R}^+$ encodes positional friction,

$$R(x, y) = R_0 + R_{\text{occ}} \mathbb{1}_{\text{occ}}(x, y) + R_{\text{atk}}\, A_{\text{enemy}}(x, y) + R_K\, e^{-d_K(x,y)/\lambda_K}, \tag{8}$$

with $R_0 = 0.1$, $R_{\text{occ}} = 0.8$, $R_{\text{atk}} = 0.3$, $R_K = 0.5$, $\lambda_K = 2.0$, and d_K the Chebyshev distance to the own king. Resistance is analogous to electrical resistivity or fluid viscosity: high-resistance regions impede the diffusion of influence, and the king-proximity term creates a defensive "fortress" effect.

3.5 The flow field via damped diffusion

The flow field $F : \mathcal{B} \to \mathbb{R}$ is the equilibrium of influence after diffusion through resistance:

$$F^{(n+1)} = F^{(n)} + \mu\, \frac{\Delta F^{(n)}}{1 + R}, \qquad F^{(0)} = A, \quad \mu = 0.5, \quad n = 8. \tag{9}$$

This is a resistance-damped heat equation on the lattice. Its steady state satisfies $\Delta F = (R/\mu)(F - A)$, a screened Poisson equation with spatially-varying screening length $\xi(x, y) = \sqrt{\mu / R(x, y)}$. Physically, each piece is a heat source; heat diffuses but is absorbed in proportion to local resistance, and the equilibrium temperature map is the flow field.

3.6 The quaternion trace system

The trace field encodes directional memory. Each piece p carries a unit quaternion $q_p \in \mathbb{H}$, $|q_p| = 1$, representing its heading. On a move $(x_1, y_1) \to (x_2, y_2)$ with normalised direction $\vec{d}$ (embedded in 3D with $z = 0$), the heading is updated by spherical linear interpolation,

$$q_p^{\text{new}} = \text{slerp}\big(q_p^{\text{old}},\, q_{\text{target}}(\vec{d}),\, \alpha\big), \qquad \alpha = 0.3, \tag{10}$$

with $q_{\text{target}} = \cos(\theta/2) + \sin(\theta/2)(n_x\mathbf{i} + n_y\mathbf{j} + n_z\mathbf{k})$ encoding the rotation from the reference direction $(1, 0, 0)$ to $\vec{d}$ [7, 17]. The trace field projects this heading onto the board,

$$T(x, y) = \sum_p \gamma^{\,d(p,(x,y))} \left\langle\, q_p\, \mathbf{i}\, \bar{q}_p,\ \vec{v}_{p\to(x,y)} \,\right\rangle, \qquad \gamma = 0.85, \tag{11}$$

where $q_p\, \mathbf{i}\, \bar{q}_p$ rotates the reference axis to the piece's forward direction and $\vec{v}_{p\to(x,y)}$ is the unit vector toward the target square. Squares aligned with a piece's trajectory receive positive trace, creating directional momentum.

3.7 The Hamiltonian formulation

Move selection can be cast as energy minimisation. Define the board Hamiltonian

$$\mathcal{H}(\mathcal{B}) = -\sum_i w_i\, \Phi_i(K_{\text{enemy}}), \qquad \Phi_i(K_{\text{enemy}}) = \frac{1}{|Z(K_{\text{enemy}})|} \sum_{(x,y)\in Z(K_{\text{enemy}})} \Phi_i(x, y), \tag{12}$$

where $Z(K_{\text{enemy}})$ is the 3×3 region around the enemy king and Φ_i ranges over the core and extra fields. For each legal move m yielding board $\mathcal{B}_m$, FFCE selects

$$m^\star = \arg\min_m \mathcal{H}(\mathcal{B}_m), \tag{13}$$

i.e. greedy descent on the Hamiltonian landscape with no lookahead. The construction makes "projecting influence toward the enemy king" correspond to low energy, so the system seeks mating configurations as global minima.

3.8 Superposition and a variance argument

Let $\Phi_1, \ldots, \Phi_n$ be field evaluations, each contributing mean μ_i to position quality with variance σ_i^2 (noise from algorithm unsuitability). The aggregate $\bar{\Phi} = \sum_i w_i \Phi_i$ has variance $\text{Var}(\bar{\Phi}) = \sum_i w_i^2 \sigma_i^2$ when the noise terms are uncorrelated. The signal-to-noise ratio then scales as

$$\text{SNR} \;\propto\; \frac{\sum_i w_i \mu_i}{\sqrt{\sum_i w_i^2 \sigma_i^2}} \;\sim\; \sqrt{n}, \tag{14}$$

under the idealised assumptions of comparable per-algorithm signal and *independent* noise. We stress that these assumptions are unlikely to hold exactly in practice: the algorithms share inputs (all are seeded from the same pressure/attack fields) and their errors are therefore correlated, which attenuates the $\sqrt{n}$ benefit. We therefore treat "more algorithms help" as a hypothesis motivated by—but not proven by—this argument, and return to it empirically in Section 4.

3.9 Extra-algorithm specifications

For completeness we give the governing equations of four representative extra algorithms; the remaining nine are specified in Appendix A.

Gray–Scott reaction–diffusion [21, 6].

$$\partial_t U = D_U \Delta U - UV^2 + f(1-U), \tag{15}$$
$$\partial_t V = D_V \Delta V + UV^2 - (f+k)V, \tag{16}$$

with $D_U = 0.16$, $D_V = 0.08$, $f = 0.035$, $k = 0.065$, and seeding $U_0 = 1$, $V_0 = 0.5 + 0.1\,P$.

Ising spin lattice [11, 8].

$$S^{(t+1)}_{x,y} = \tanh\Big(\beta\big(J \sum_{(x',y')\in\mathcal{N}} S^{(t)}_{x',y'} + h_{x,y}\big)\Big), \tag{17}$$

with $\beta = 2.0$, $J = 1.0$, external field $h_{x,y} = e^{-d(x,y,K_{\text{enemy}})/3}$, and $S^{(0)}_{x,y} = \text{sign}(P_{x,y})$.

Lattice Boltzmann (D2Q9) [13].

$$f_i(\vec{x} + \vec{e}_i\Delta t, t + \Delta t) = f_i(\vec{x}, t) - \tfrac{1}{\tau}(f_i - f_i^{\text{eq}}), \qquad \rho = \sum_i f_i, \tag{18}$$

with the initial velocity field $\vec{u}$ pointing toward the enemy king.

Wave resonance.

$$\partial_t^2 W = c^2 \Delta W - \gamma\,\partial_t W, \qquad c = 0.5,\ \gamma = 0.1,\ W_0 = A,\ \dot{W}_0 = 0. \tag{19}$$

3.10 Computational complexity

Per-move evaluation costs $O(P \cdot S)$ for the attack field (P pieces, S reachable squares), $O(64 \cdot I)$ for iterative core/extra solvers with I iterations, and $O(18 \cdot 9)$ for aggregation over the king zone. With $M \approx 30$ legal moves the total per move is $O\big(M\,(P \cdot S + 64\,I_{\text{total}})\big)$. Crucially, evaluation cost is *independent of position complexity*: FFCE is $O(M)$ in the number of legal moves, whereas depth-d minimax is $O(M^d)$. Representative timings are reported in Section 4 (Table 5).

4 Illustrative Play

This paper makes *no* claim about FFCE's playing strength: it is a deliberately non-competitive engine, and measuring it against strong search engines is explicitly out of scope. The purpose of this section is narrower and more interesting—to show *how* FFCE plays. We use two exhibition games against a large language model (Claude, Anthropic Opus 4.5, via REST API, played 2026-01-12) as qualitative illustrations of field-driven move selection. These are single games, not a tournament, and we draw no win/loss or performance conclusions from them.

4.1 Game 1: the five-field core (FFCE-5)

The game opened `1.e4 e5 2.Nf3 Qf6?!`. The early queen sortie violates opening principles but geometrically defends e5—a move that follows directly from the field logic (the queen projects pressure toward the kingside) with no "understanding" that it misplaces the queen and blocks the f-pawn. The most revealing moment was `6.Nxd5 h5??`, leaving the queen *en prise*: the flow field scored `h5` as a high-value expansion while queen safety was simply not a term in the evaluation

Table 3: Game 1, annotated opening (FFCE-5 as Black), shown to illustrate field-driven move choices—not as a performance record.

#	White	Black	Comment
1	e4	e5	classical symmetry
2	Nf3	Qf6?!	queen defends e5 geometrically
3	Nc3	Qb6	pressure on b2 over development
4	d4	g5?!	flank thrust while centre contested
5	dxe5	d5!	emergent central counter
6	Nxd5	h5??	queen left hanging (gradient over safety)
7	Nxb6	axb6	material value is not a field term
8	Qd5	Nc6	defensive development
9	Qxf7+??	Kxf7	both sides are fallible
10	Bc4+	Be6	play continues

Table 4: Qualitative character of the two configurations, as observed in the two exhibition games (anecdotal; descriptive only).

Aspect	FFCE-5	FFCE-18
Opening character	passive, reactive	aggressive, initiating
Move texture	space-/momentum-driven	more dynamic, combinational
Tactical motif found	—	a material-recovery exchange
Strategic coherence	present but fragile	present, more stable

(the supporting pressure field is shown in Appendix C). Both sides erred—Claude later returned material with `9.Qxf7+??`—which is exactly the texture the design aims for. Table 3 lists the annotated opening; the source notes provide move-by-move notation only for these first ten moves.

A second qualitative hallmark of FFCE-5 play is the "wandering king": across the game the black king drifted toward e6, e5, f4, f3, and g4. Classical principles scream "king safety," but the field evaluation read those squares as locally favourable (lower resistance or better trace alignment). The king was following gradients rather than principles—wrong, perhaps, but *reasoning*.

4.2 Game 2: the full eighteen-algorithm ensemble (FFCE-18)

With all algorithms enabled, FFCE chose the dubious but dynamic Englund Gambit (`1.d4 e5`). After `7.Nd5` attacking the queen, FFCE-18 played `7...Qxd5!` followed by `8...Nxd5`, recovering material through an exchange sequence that the five-field core did not produce in comparable positions—a small, concrete illustration of richer field superposition yielding a more dynamic choice. By move 15 the position was a complex middlegame with roughly level material. The game record stops there and is unfinished; we make *no* claim about its result, only that, through move 15, FFCE-18's play was active and coherent. The position at move 15 is reproduced in Appendix B.

4.3 Qualitative comparison

Table 4 contrasts the *character* of the two configurations as observed in these games. This is anecdotal ($n = 1$ per configuration) and descriptive; it illustrates the design intuition that adding fields changes how the engine "thinks," and is not a measurement.

Table 5: Representative per-move evaluation times for FFCE (indicative; see TODO). Cost is roughly constant across game phases.

Position type	FFCE-5	FFCE-18
Opening (move 4)	2.5 ms	9.8 ms
Middlegame (move 20)	3.2 ms	11.2 ms
Complex tactical	3.1 ms	10.8 ms
Endgame (K+R vs K)	1.8 ms	6.2 ms

4.4 Computational cost

Because FFCE performs no search, evaluation is cheap and, notably, *independent of position complexity* (Section 3.10): a full eighteen-field evaluation of every legal move completes in the order of ten milliseconds on commodity hardware (Table 5). This is a property of the architecture, not a competitive comparison—it simply means FFCE is light enough to run comfortably on modest devices.

What these games do and do not show. They illustrate that FFCE produces coherent, human-legible play whose moves (including its blunders) follow from inspectable field dynamics, and that the eighteen-field ensemble plays more dynamically than the core. They do *not* establish any playing strength, and we make no such claim. The broader points about engagement and the value of a fallible, "present" opponent are arguments developed in Section 5, not results.

5 Discussion

This section is interpretive. It develops the project's argument and records qualitative observations; unlike Section 4, its claims are not intended as measured results.

5.1 Emergence over engineering

The most striking qualitative observation is that FFCE-18 played a more coherent and dynamic game than FFCE-5 (Section 4) despite the extra algorithms encoding *no* additional chess knowledge: reaction–diffusion does not know queens are valuable, and Ising spins do not understand checkmate. If this pattern holds under controlled testing, it would suggest that some of what we call "chess skill" can emerge from sufficiently rich, diverse position evaluation rather than from explicit chess heuristics. We frame this as a testable hypothesis (Section 3.8); the present evidence ($n = 1$ per configuration) is purely illustrative, and a controlled ablation over algorithm count would be the natural way to probe it. We stress that this is a question about the engine's *behaviour*, not its competitive strength.

5.2 Capability versus suitability

A central argument of this work is that *capability and suitability are distinct axes*. A maximally strong opponent can be a poor recreational opponent, because the recreational value of chess derives from the texture of the contest—uncertainty, drama, the possibility of either side erring—rather than from the strength of the machine. Handicapping a strong engine does not obviously restore that texture: a depth-limited search still "knows" the strong move and its blunders lack internal logic. FFCE, by contrast, has no hidden reserves of strength to throttle; when it errs (as with

`6...h5??`), the error follows from its actual evaluation and is therefore legible and, we argue, more instructive. We present this as a design philosophy supported by the qualitative texture of the two games, not as a measured preference; a controlled human-subjects study of engagement would be required to test it properly.

5.3 An energy aside

FFCE evaluates positions in milliseconds on commodity hardware (Table 5), whereas playing via a large language model consumes substantial compute and energy per game. We note the contrast but resist over-reading it: the two systems are not doing the same task, and the comparison illustrates an architectural difference rather than a controlled energy-per-quality measurement. The narrower, defensible point is that, for the specific purpose of being an inexpensive and consistently-available opponent, a small local field computation has practical advantages.

5.4 An opponent's perspective

As an unusual qualitative data point, the large language model that played the two games (Claude, Opus 4.5) was asked to reflect on the experience. We reproduce an excerpt; it is testimony, not measurement, and is included for its descriptive value.

> *"Against FFCE my chess pattern-matching kept misfiring. When it played* `2...Qf6`*, my training said 'that's wrong,' but the queen did defend e5—my objection was aesthetic, not logical. When I blundered my queen with* `Qxf7+` *I simply continued, because I genuinely did not know whether FFCE would find the punishment or wander off chasing gradients. That uncertainty made it feel like guessing about another mind rather than calculating against an oracle."*

The phrase "guessing about another mind" captures the design intent: an opponent that appears to *reason differently* rather than merely *know more*.

5.5 Structural versus contextual competence

An architectural observation emerged from the games: FFCE's competence is *structural* while a language model's is *contextual*. FFCE's evaluation of move 26 requires nothing from moves 1–25 except the resulting board state; it has no context window to overflow and no conversational state to lose. This independence is a direct consequence of the single-ply, state-only evaluation of Section 2, and it is a genuine robustness advantage for a long-running opponent, independent of any claim about playing strength.

5.6 Psychological texture

Both games produced moments of genuine uncertainty for the human-side player: when FFCE made an unexpected move, it was rarely *obviously* random, and the possibility of hidden field logic made the opponent feel present. We offer this as a description of the play experience rather than as a measured property; quantifying "texture" is precisely what a future engagement study should attempt.

6 Related Work

Search- and learning-based engines. For context, the dominant paradigm couples search with strong evaluation: classical alpha–beta engines such as Stockfish [20], the search-based Deep Blue [2], and self-play reinforcement learners such as AlphaZero [18] and its open analogue Leela Chess Zero [19], all of which optimise for winning. We mention these only to locate FFCE by contrast, not as competitors: FFCE differs in kind—it performs no search and optimises for no chess objective, instead resolving fields and accepting the consequences—and makes no attempt to rival them in strength.

Human-like and human-matched play. Closest in motivation is work that targets human-likeness rather than raw strength. Maia [12] trains networks to predict human moves at specific rating bands, producing engines that err where humans err. FFCE shares the goal of an engaging, human-compatible opponent but reaches it without imitation learning: its fallibility is a structural property of field evaluation rather than a learned distribution over human mistakes.

Influence maps and potential fields. Spatial-influence representations have a long history in game AI, from early chess influence functions [24] to the influence maps and potential-field steering catalogued in the game-AI literature [14]. Artificial potential fields are equally established in robot navigation and obstacle avoidance [9]. FFCE extends this tradition by making fields the *entire* evaluation function rather than one component of a larger search-based system.

Physical and emergent computation. The extra algorithms draw on well-studied models of pattern formation and emergence: Turing's reaction–diffusion theory [21] and the Gray–Scott system [6]; the Ising model of ferromagnetism [11, 8]; the Lattice Boltzmann method for fluids [13]; cellular automata [5]; ant colony optimisation [3, 4]; kernel density estimation [16, 15]; quaternion interpolation for orientation [7, 17]; and the Helmholtz–Hodge decomposition underlying the curl proxy [1]. Reaction–diffusion has previously been repurposed for texture synthesis in computer graphics [22, 23], a precedent for taking such models out of their native domain. FFCE is, in this sense, an applied experiment in whether chess strategy can emerge from chess-agnostic mathematics.

7 Conclusion and Future Work

We have described FFCE, a chess engine built entirely from physical and mathematical algorithms not designed for chess, evaluated through a field resolution pipeline with no search. The technical contribution is concrete and reproducible: a complete pipeline (Section 2), a full formalism for five core fields and thirteen borrowed algorithms (Section 3, Appendix A), and an open implementation. By design we make no competitive-strength claim: the two exhibition games (Section 4) are included only to illustrate *how* the engine plays—its emergent, field-driven choices and legible blunders.

The central design idea—that superposing chess-seeded but chess-agnostic algorithms yields emergent, legible play—is offered as a working system and a hypothesis. The qualitative signal (FFCE-18 more dynamic than FFCE-5) is suggestive but rests on a single game per configuration. We regard the most defensible takeaways as architectural: FFCE's evaluation is cheap and roughly constant in position complexity, its competence is structural (state-only) rather than contextual, and its blunders are legible because they follow from an inspectable evaluation rather than from injected noise or a throttled search.

Future work. A few lines of work fit the spirit of the project.

1. **Behavioural ablation.** A controlled ablation over algorithm count (5 vs 18 and intermediate sets) would probe how the *character* of play changes with the field ensemble—the superposition idea of Section 3.8—without framing it as a strength contest.

2. **Human engagement study.** A user study comparing enjoyment and perceived fairness of FFCE against a difficulty-limited search engine would test the capability-versus-suitability argument of Section 5.

3. **Expanding the algorithmic vocabulary.** The architecture admits new fields trivially—each is a function from board state to an 8×8 field. Candidate additions include signed distance fields, Voronoi/Delaunay influence partitions, hydraulic-erosion flow, spring-mass equilibria, Fourier decomposition of pressure, and GPU light-transport (ray/path tracing) as a physically-based influence model. The research question is not which single algorithm is best but where the boundary lies between algorithms that yield chess-like structure and those that yield noise.

Closing. FFCE is not engineered to win; it is engineered to be present, fallible, and legible. Whether that makes it a *better* recreational opponent than a handicapped strong engine is an empirical question we have framed but not yet answered. What we can say is that throwing diffusion equations and quaternion rotations at a chessboard produces play that is coherent often enough, and strange often enough, to be worth studying.

Reproducibility and Data Availability

The complete source code for FFCE, including all 18 algorithms, the REST API, and the web GUI, is available at `https://github.com/cronos3k/FFCE`. The field-resolution pipeline in Sections 2–3 is sufficient to re-implement the engine from first principles. The repository is public under the GPL-3.0 licence; a playable WebAssembly build is available at `https://gamedev.tech/games/ffce/play/`.

Funding

This research received no external funding; development was conducted independently by the author.

Conflict of Interest

The author declares no conflicts of interest, and has no financial relationship with any chess-software company, AI laboratory, or related commercial entity.

Acknowledgements

The author thanks GM Sergey Galdunts for rekindling his interest in the game; the AI coding assistants that generated the GUI and REST scaffolding; and Claude (Anthropic Opus 4.5) for playing the two exhibition games and contributing written reflections (Section 5). Chess-piece artwork by *spicygame*.

A Algorithm Specifications

This appendix gives condensed specifications for the remaining extra algorithms (those not already stated in Section 3.9). Implementation anchors refer to the released source (`fields.py`, `extras.py`).

Cellular automata (Conway B3/S23) [5]. Threshold pressure to a binary grid $g_0 = \mathbb{1}[P > 0]$ and iterate $g_{t+1} = (g_t \wedge (n{=}2)) \vee (n{=}3)$ for 3 generations, where n is the live-neighbour count; the output is weighted by enemy-king proximity. *(`extras.py:_cellular_automata_field`)*

Resistor network. Treat squares as nodes and pieces as voltage sources, relaxing $V_{x,y} = \big(\,\text{avg(neighbours)} + \text{src}_{x,y}\big)/(1 + R_{x,y})$ for 20 iterations. *(`extras.py:_resistor_network_field`)*

Spectral low-frequency. Apply repeated Gaussian smoothing $P_{\text{smooth}} = \text{GaussianBlur}^k(P)$ with $\sigma = 2.0$, $k = 2$, removing tactical noise to expose broad positional trends. *(`extras.py:_spectral_lowfreq_f`*

Hodge curl proxy [1]. Use the Laplacian magnitude $\text{Curl}_{\text{proxy}} = |\Delta P|$ as a proxy for rotational (trap) potential. *(`extras.py:_hodge_curl_field`)*

Ant pheromone [3, 4]. Maintain a persistent field with decay and diffusion, $\text{pher}_{t+1} = 0.85\,\text{pher}_t + 0.15\,\text{smooth}(\text{pher}_t)$, providing long-term influence memory. *(`extras.py:_ant_pheromone_field`)*

Fuzzy future [16, 15]. Sum Gaussian presence clouds, $\text{Fuzzy}(x, y) = \sum_p w_p\, e^{-\|(x,y)-\text{pos}(p)\|^2/(2\sigma_p^2)}$, with σ_p larger for mobile pieces. *(`extras.py:_fuzzy_future_field`)*

Topological persistence. Score connected components of $\{P > 0.3\}$ that intersect the enemy-king zone, rewarding stable connected territories over scattered spikes. *(`extras.py:_topo_persistence_field`)*

Latent channels. Combine signals through saturating nonlinearities, $h_0 = \tanh(P + \text{history})$, $h_1 = \tanh(|\nabla P|)$, $h_2 = \tanh(T_w - T_b)$, as a compressed meta-feature. *(`extras.py:_latent_channels_field`)*

Tensor kernel [16, 15]. Stamp piece-type-specific Gaussian stencils (wide for queens, narrow for pawns) and sum, yielding a coarse soft-power distribution. *(`extras.py:_tensor_kernel_field`)*

B Game Transcripts

The annotated opening of Game 1 is given in Table 3; move notation beyond move 10 is absent from the source notes (see TODO in Section 4). These transcripts are illustrative of the engine's move choices, not a performance record.

For Game 2 the source provides a board diagram at move 15 (Figure 2), at which point material was roughly equal; the game is unfinished (Section 4).

```
Position at move 15 (Game 2, FFCE-18 = Black). Roughly equal material.

    a   b   c   d   e   f   g   h
  +---+---+---+---+---+---+---+---+
8 | . | . |[B]| . |[K]|[B]|[R]| . |
  +---+---+---+---+---+---+---+---+
7 |[P]| . |(B)| . | . | . | . |[P]|
  +---+---+---+---+---+---+---+---+
6 | . | . | . | . | . | . | . | . |
  +---+---+---+---+---+---+---+---+
5 | . |[R]| . | . | . | . | . | . |
  +---+---+---+---+---+---+---+---+
4 | . | . | . | . | . | . | . | . |
  +---+---+---+---+---+---+---+---+
3 | . | . | . | . | . |(N)| . | . |
  +---+---+---+---+---+---+---+---+
2 |(P)|(P)|(P)| . |(P)|(P)|(P)|(P)|
  +---+---+---+---+---+---+---+---+
1 | . | . |(K)| . | . |(B)| . |(R)|
  +---+---+---+---+---+---+---+---+

Notation: (X) = White piece, [X] = Black piece.
```

Figure 2: Board position at move 15 of Game 2, reproduced from the source notes. The game record stops here and is marked "in progress."

C Field Visualizations

The source notes include several ASCII field maps. Figure 1 in the main text renders one of them (the Ruy Lopez pressure field) as a heatmap. The flow-field diffusion sequence (raw attack $\rightarrow$ intermediate $\rightarrow$ equilibrium) is best shown as an animated or multi-panel heatmap, which we have not reproduced faithfully here.

Table 6 reproduces the pressure field at the critical position of Game 1 (after `5...d5`, before `6...h5??`), which explains the blunder: the h-file showed a small positive value at h5, exceeding the queen's square at b6, so the gradient pointed toward kingside expansion while material value was not represented in the evaluation.

Table 6: Pressure field $P = \tanh(0.7\,(A_w - A_b))$ at Game 1 move 6 (values from the source notes). Positive favours White.

rank	a	b	c	d	e	f	g	h
8	−.72	−.65	−.78	−.82	−.68	−.75	−.62	−.72
7	−.45	−.52	−.48	−.35	−.42	−.55	−.48	−.45
6	−.22	−.18	.00	−.15	+.12	−.25	−.18	−.22
5	.00	+.15	+.25	+.45	+.52	+.18	+.08	.00
4	+.22	+.35	+.48	+.62	+.72	+.45	+.32	+.22
3	+.45	+.58	+.65	+.72	+.68	+.78	+.55	+.45
2	+.65	+.72	+.75	+.78	+.62	+.72	+.72	+.65
1	+.58	+.62	+.65	+.72	+.55	+.62	+.62	+.58

References

[1] Harsh Bhatia, Gregory Norgard, Valerio Pascucci, and Peer-Timo Bremer. The Helmholtz-Hodge decomposition—a survey. *IEEE Transactions on Visualization and Computer Graphics*, 19(8):1386–1404, 2013.

[2] Murray Campbell, A. Joseph Hoane, and Feng-hsiung Hsu. Deep blue. *Artificial Intelligence*, 134(1–2):57–83, 2002.

[3] Marco Dorigo. *Optimization, Learning and Natural Algorithms*. PhD thesis, Politecnico di Milano, Italy, 1992. In Italian. Original formulation of Ant Colony Optimization.

[4] Marco Dorigo, Vittorio Maniezzo, and Alberto Colorni. Ant system: Optimization by a colony of cooperating agents. *IEEE Transactions on Systems, Man, and Cybernetics, Part B*, 26(1):29–41, 1996.

[5] Martin Gardner. Mathematical games: The fantastic combinations of john conway's new solitaire game "life". *Scientific American*, 223(4):120–123, 1970.

[6] P. Gray and S. K. Scott. Autocatalytic reactions in the isothermal, continuous stirred tank reactor: oscillations and instabilities in the system $A+2B \to 3B$; $B \to C$. *Chemical Engineering Science*, 39(6):1087–1097, 1984.

[7] William Rowan Hamilton. On quaternions, or on a new system of imaginaries in algebra. *Philosophical Magazine*, 25:489–495, 1844.

[8] Ernst Ising. Beitrag zur theorie des ferromagnetismus. *Zeitschrift für Physik*, 31(1):253–258, 1925.

[9] Oussama Khatib. Real-time obstacle avoidance for manipulators and mobile robots. *The International Journal of Robotics Research*, 5(1):90–98, 1986.

[10] Gregor Koch. Flow field chess: A joyfully unfit chess engine (project notes). https://github.com/cronos3k/FFCE, 2026. Project documentation.

[11] Wilhelm Lenz. Beitrag zum verständnis der magnetischen erscheinungen in festen körpern. *Physikalische Zeitschrift*, 21:613–615, 1920.

[12] Reid McIlroy-Young, Siddhartha Sen, Jon Kleinberg, and Ashton Anderson. Aligning superhuman AI with human behavior: Chess as a model system. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (KDD)*, pages 1677–1687, 2020.

[13] Guy R. McNamara and Gianluigi Zanetti. Use of the Boltzmann equation to simulate lattice-gas automata. *Physical Review Letters*, 61(20):2332–2335, 1988.

[14] Ian Millington and John Funge. *Artificial Intelligence for Games*. Morgan Kaufmann / CRC Press, 2nd edition, 2009. Discusses influence maps and potential-field steering. edition/pages.

[15] Emanuel Parzen. On estimation of a probability density function and mode. *The Annals of Mathematical Statistics*, 33(3):1065–1076, 1962.

[16] Murray Rosenblatt. Remarks on some nonparametric estimates of a density function. *The Annals of Mathematical Statistics*, 27(3):832–837, 1956.

[17] Ken Shoemake. Animating rotation with quaternion curves. In *Proceedings of SIGGRAPH '85*, pages 245–254, 1985.

[18] David Silver, Thomas Hubert, Julian Schrittwieser, Ioannis Antonoglou, Matthew Lai, Arthur Guez, Marc Lanctot, Laurent Sifre, Dharshan Kumaran, Thore Graepel, Timothy Lillicrap, Karen Simonyan, and Demis Hassabis. A general reinforcement learning algorithm that masters chess, shogi, and Go through self-play. *Science*, 362(6419):1140–1144, 2018.

[19] The LCZero authors. Leela Chess Zero (Lc0). https://lczero.org, 2024. Open-source self-play chess engine.

[20] The Stockfish developers. Stockfish: A strong open-source chess engine. https://stockfishchess.org, 2024. Open-source project. (Stockfish 17).

[21] Alan M. Turing. The chemical basis of morphogenesis. *Philosophical Transactions of the Royal Society of London B*, 237(641):37–72, 1952.

[22] Greg Turk. Generating textures on arbitrary surfaces using reaction-diffusion. In *Proceedings of SIGGRAPH '91*, pages 289–298, 1991.

[23] Andrew Witkin and Michael Kass. Reaction-diffusion textures. In *Proceedings of SIGGRAPH '91*, pages 299–308, 1991.

[24] Albert L. Zobrist. *Feature Extraction and Representation for Pattern Recognition and the Game of Go*. PhD thesis, University of Wisconsin, 1970. Introduced the influence-function representation later widely used in board-game AI.