Eur. Phys. J. B (2022) 95:116
https://doi.org/10.1140/epjb/s10051-022-00378-x

THE EUROPEAN PHYSICAL JOURNAL B

Regular Article - Computational Methods

Better integrators for functional renormalization group calculations

Jacob Beyer[1,2,3,a], Florian Goth[1,b], and Tobias Müller[1,c]

[1] Institute for Theoretical Physics, University of Würzburg, Würzburg 97074, Germany
[2] Institute for Theoretical Solid State Physics, RWTH Aachen University, Aachen 52056, Germany
[3] School of Physics, University of Melbourne, Parkville VIC 3010, Australia

Received 29 March 2022 / Accepted 5 July 2022 / Published online 25 July 2022

Abstract. We analyze a variety of integration schemes for the momentum space functional renormalization group calculation with the goal of finding an optimized scheme. Using the square lattice $t - t'$ Hubbard model as a testbed we define and benchmark the quality. Most notably we define an error estimate of the solution for the ordinary differential equation circumventing the issues introduced by the divergences at the end of the FRG flow. Using this measure to control for accuracy we find a threefold reduction in number of required integration steps achievable by choice of integrator. We herewith publish a set of recommended choices for the functional renormalization group, shown to decrease the computational cost for FRG calculations and representing a valuable basis for further investigations.

1 Introduction

The functional renormalization group (FRG) has proven itself to be a versatile theoretical framework to study competing ordering tendencies and other many-body phenomena both in itinerant fermionic [1–3] and quantum spin systems [4–8]. The underlying flow equations are a large system of coupled, non-linear ordinary differential equations (ODEs), the solution of which stipulated the development of highly sophisticated numerical codebases [9–11]. So far, algorithmical improvements mainly focussed on the efficient representation of vertex functions and optimizing momentum integrations on the right-hand-side (RHS) of the flow equations, with the numerical solution of the ODE itself only treated as a necessity rather than an opportunity for improvements. Consequently, a simple Euler step was used in the majority of FRG calculations to date, see e.g. [3,4,10], with the inception of higher-order solvers from the Runge–Kutta family being a fairly recent development in the field [9,11–21].

The appropriate choice of integrator will lead to improved accuracy at the same number of evaluations of the RHS, which constitutes the main computational bottleneck of any FRG calculation. Alternatively, one can significantly reduce this number while maintaining good accuracy. In contrast to many other optimizations, this gain does not introduce any further physical approximations, but solely rests in mathematics. As a prototypical setup to study the influence of different integrators, we choose the momentum space FRG in the simplest truncation scheme, only considering the four-particle vertex $\Gamma^{(4)}$ while neglecting self-energy effects. Additionally, we limit ourselves to the well-understood case of the square lattice Hubbard model with hopping up to second-nearest neighbors [9,14,15,22–34].

We explicitly refrain from implementing more sophisticated approximation schemes [9,14,15,30,35,36] or multi-band extensions [3,37–41], not only for clarity of analysis, but rather focus on benchmarking a large variety of different standard ODE solvers, most available in existing libraries [42,43]. Nevertheless, we expect our conclusions to provide a starting ground for the application of better integrators to more sophisticated physical approximations.

The paper is organized as follows: after an introduction of our notation of momentum space FRG and the approximations employed in our implementation in Sect. 2.1, we provide an overview over various integration schemes in Section 2.2 and define tangible metrics to judge the quality of the different schemes in Sect. 2.3. In Sect. 3 we employ a two-stage elimination procedure. In a first step we sort out all integrators incapable of reproducing the physical ordering instability for the nearest-neighbor only model. We subsequently analyze the remaining set over a larger parameter space including second-neighbor hopping to distinguish optimal choices.

Jacob Beyer, Florian Goth and Tobias Müller contributed equally to this work.

[a] e-mail: beyer@physik.rwth-aachen.de
[b] e-mail: florian.goth@physik.uni-wuerzburg.de
[c] e-mail: tobias.mueller@physik.uni-wuerzburg.de (corresponding author)

2 Methods

2.1 FRG in brevity

The following discussion of FRG will be reduced to an absolute minimum and only serves as a means to introduce the notation used. For detailed derivations we refer the interested reader to standard literature [1–3].

In the following, we will focus solely on the square-lattice $t-t'$ Hubbard model defined by

$$H_0 = -\sum_{\langle i,j\rangle,\sigma} t\, c^\dagger_{i,\sigma} c^\dagger_{j,\sigma} - \sum_{\langle\langle i,j\rangle\rangle,\sigma} t'\, c^\dagger_{i,\sigma} c^\dagger_{j,\sigma} - \sum_{i,\sigma} \mu\, c^\dagger_{i,\sigma} c^\dagger_{i,\sigma}\,, \tag{1}$$

$$H_\mathrm{I} = U \sum_{i,\sigma} n_{i,\sigma} n_{i,\bar\sigma}\,, \tag{2}$$

where t (t') denote (next-) nearest-neighbor hopping amplitudes, μ is the chemical potential and U the strength of the onsite Hubbard interaction. Due to the spin-rotation invariance of Eqs. (1) and (2), we can constrain our derivation to the SU(2)-symmetric set of equations. To simplify the functional dependence of the FRG equations, we invoke the static approximation of the problem, i.e. consider only the zero Matsubara frequency component of the vertex functions, and neglect self-energy effects [3,25]. We furthermore calculate at zero temperature to enable analytic evaluation of all occurring Matsubara frequency summations.

The SU(2)-reduced FRG equation in the conventional truncation scheme including contributions up to the four-point vertex $\Gamma^{(4)}$ is diagrammatically shown in Fig. 1 and can be expanded to (using V as SU(2)-symmetrized four-point vertex):

$$\begin{aligned}\frac{\mathrm{d}}{\mathrm{d}\Lambda} V_{\boldsymbol{k}_0,\boldsymbol{k}_1,\boldsymbol{k}_2} = \;& V_{\boldsymbol{k}_0,\boldsymbol{k}_1,\boldsymbol{l}}\, \dot L^{-,\Lambda}_{\boldsymbol{l},-\boldsymbol{l}+\boldsymbol{k}_0+\boldsymbol{k}_1} V_{\boldsymbol{l},-\boldsymbol{l}+\boldsymbol{k}_0+\boldsymbol{k}_1,\boldsymbol{k}_2}\\ &+ V_{\boldsymbol{k}_0,\boldsymbol{l}-\boldsymbol{k}_0+\boldsymbol{k}_2,\boldsymbol{k}_2}\, \dot L^{+,\Lambda}_{\boldsymbol{l},\boldsymbol{l}-\boldsymbol{k}_0+\boldsymbol{k}_2} V_{\boldsymbol{l},\boldsymbol{k}_1,\boldsymbol{l}-\boldsymbol{k}_0+\boldsymbol{k}_2}\\ &+ V_{\boldsymbol{k}_0,\boldsymbol{l}-\boldsymbol{k}_0+\boldsymbol{k}_2,\boldsymbol{k}_2}\, \dot L^{+,\Lambda}_{\boldsymbol{l},\boldsymbol{l}-\boldsymbol{k}_0+\boldsymbol{k}_2} V_{\boldsymbol{k}_1,\boldsymbol{l},\boldsymbol{l}-\boldsymbol{k}_0+\boldsymbol{k}_2}\\ &+ V_{\boldsymbol{k}_0,\boldsymbol{l}-\boldsymbol{k}_0+\boldsymbol{k}_2,\boldsymbol{l}}\, \dot L^{+,\Lambda}_{\boldsymbol{l},\boldsymbol{l}-\boldsymbol{k}_0+\boldsymbol{k}_2} V_{\boldsymbol{l},\boldsymbol{k}_1,\boldsymbol{l}-\boldsymbol{k}_0+\boldsymbol{k}_2}\\ &+ V_{\boldsymbol{k}_0,\boldsymbol{l}-\boldsymbol{k}_1+\boldsymbol{k}_2,\boldsymbol{l}}\, \dot L^{+,\Lambda}_{\boldsymbol{l},\boldsymbol{l}+\boldsymbol{k}_1-\boldsymbol{k}_2} V_{\boldsymbol{l},\boldsymbol{k}_1,\boldsymbol{k}_2}\,,\end{aligned} \tag{3}$$

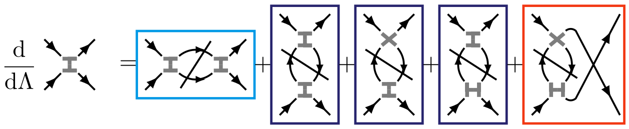

Fig. 1 Diagrammatic representation of the functional renormalization group equations. We show the SU(2)-symmetric flow equation, where the SU(2)-degree of freedom is kept constant along the lines of the vertex. The line through the loop represents the scale derivative d/dΛ. We have indicated the three channels of the FRG flow, differing in the bosonic transfer momentum: P-, C-, and D-channel

where we suppress the, in the SU(2) Hubbard model, unnecessary spin indices. We also use a modified Einstein sum convention to indicate the momentum-integration in $\boldsymbol{l}$ over the Brillouin zone (BZ). To treat the continuous momentum-dependence of the vertex function, we employ the "grid-FRG" momentum discretization scheme in the BZ with the additional refinement-scheme described in Ref. [44]. Eq. (3) will commonly be referred to as the right-hand side (RHS) of the FRG equation, which we need to compute at each flow step iteration.

The $\dot L^\Lambda$ given in the RHS equation is the derivative w.r.t. the scale Λ of the regulated loop

$$L^{\pm,\Lambda} = \sum_n G^\Lambda(\boldsymbol{k}, \mathrm{i}\omega_n) G^\Lambda(\boldsymbol{k}', \pm\mathrm{i}\omega_n)\,. \tag{4}$$

The FRG regulator $\Theta(\Lambda)$ here is introduced multiplicatively by defining $G^\Lambda = \Theta(\Lambda) G$ where G is the bare propagator. We have used both the particle-particle loop $L^{-,\Lambda}$ and the particle-hole loop $L^{+,\Lambda}$ in our notation. As a regulator we choose the Ω-cutoff $\Theta(\Lambda) = \frac{\omega^2}{\omega^2-\Lambda^2}$ [29] with high numerical stability. The analytic Matsubara frequency summations then yield:

$$\dot L^{\pm,\Lambda}_{\boldsymbol{l},\boldsymbol{l}'} = \frac{\mathrm{d}}{\mathrm{d}\Lambda} L^{\pm,\Lambda}_{\boldsymbol{l},\boldsymbol{l}'} \tag{5}$$

$$= \begin{cases} \dfrac{\pm 1/4}{\epsilon \mp \epsilon'} \left(\dfrac{\epsilon(3|\epsilon| + \Lambda)}{(|\epsilon| + \Lambda)^3} \mp \dfrac{\epsilon'(3|\epsilon'| + \Lambda)}{(|\epsilon'| + \Lambda)^3} \right) \\ \mp \dfrac{3\epsilon^2 - 4|\epsilon|\Lambda - \Lambda^2}{4(|\epsilon| + \Lambda)^4} \qquad \text{if } \epsilon = \epsilon'\,, \end{cases} \tag{6}$$

with $\epsilon = \epsilon(l)$ and $\epsilon' = \epsilon(l')$ being the dispersion ϵ of the non-interacting Hamiltonian H_0 for specific momenta l, l'. To obtain the effective low-energy vertex we integrate the differential equation (RHS) starting at infinite (i.e. large compared to bandwidth) $\Lambda = \Lambda_\infty$ and approach $\Lambda \to 0$. When encountering a phase transition the loop derivatives will diverge in conjunction with the associated susceptibilities and we terminate the flow. This is the integration for which we attempt to find an optimized solver. Due to the divergence we can not finalize the calculations using the solvers but instead hard-terminate them once the maximum element of the vertex exceeds a threshold V_max.

2.2 Integrators

The possible choices of integrators are plentiful but in the following we highlight some of the most prevalent algorithms as well as some which will prove to excel during our testing. The measurement schemes used to determine quality are provided at the end of this section. To numerically solve the non-linear non-autonomous differential equation

$$\frac{\mathrm{d}}{\mathrm{d}\Lambda} V = \mathrm{RHS}(V, \Lambda) \tag{7}$$

we utilize both single-step and multi-step methods [45] to obtain an iteration procedure

$$V_{n+1} = V_n + \Delta_{\Lambda,n}\Phi(\Lambda_n, V_n, \Delta_{\Lambda,n}) \tag{8}$$

that we terminate at the divergence of V.

We almost exclusively focus on integrators that are implemented in the excellent `DifferentialEquations.jl` package [42] written in the Julia programming language. We emphasize integrators that are well-known and commonly employed, such as the Runge–Kutta class and also widely available in other programming languages. A full list of the considered (including all disregarded) integrators can be found in Appedix A. We want to note here that non-adaptive integrators were eliminated as finer step sizes at lower Λ are required for efficient solution of the flow equations.

2.2.1 Single-step methods

A single-step method is characterized by utilizing at each step a starting value V_n and additional evaluations of the RHS to construct V_{n+1}. The most well-known family of methods in this class are the Runge–Kutta type integrators where the function Φ is constructed as a weighted average of evaluations of RHS within the interval $[\Lambda_n, \Lambda_{n+1}]$ such that it coincides up to the respective order with its Taylor polynomial:

$$V_{n+1} = V_n + \Delta_\Lambda \sum_{i=1}^{s} b_i k_i \tag{9}$$

$$k_i = \mathrm{RHS}\left(\Lambda_n + c_i\Delta_\Lambda, V_n + \Delta_\Lambda \sum_{j=1}^{s} a_{ij}k_j\right) \tag{10}$$

The method is explicit if $a_{ij} = 0$ for $j \geq i$ else it is an implicit method. Adaptivity can be included by a time-step dependent $\Delta_{\Lambda,n}$. The specifics of the methods are covered in tremendous detail in the literature, e.g., [45].

2.2.2 Example: adaptive explicit Euler

The conceptually simplest integrator in FRG that serves as the baseline for this work is the adaptive explicit Euler described in Algorithm 1. Note that we have included a function $f(\Lambda, V_{\max}) = \min(\max(a\Lambda/V_{\max}, \Delta_{\min}), \Delta_{\max})$ which is an adaption scheme specifically designed to reduce the step-width in the proximity of the expected flow divergence. Here a is a small number determining the relative speed of the integration while $\Delta_{\max}, \Delta_{\min}$ are chosen based on previous experience: $\Delta_{\max} \gtrsim \Lambda \times 10^{-2}$ and $\Delta_{\min} \lesssim \Lambda \times 10^{-5}$.

2.2.3 A note on implicit Runge–Kutta methods

We have attempted to include implicit integration schemes in our analysis but were inhibited by their high memory cost. While for explicit methods the requirements are understood to be $m \times$ `sizeof`(V) - m being the number of stored evaluations of RHS - for the implicit methods we must instead consider the size of the Jacobian we are attempting to invert. This is proportional to the square of the size of a single solution (as it is a linear map between two of them) and is thus $\propto$ `sizeof`$(V)^2$. Even for the relatively small scale systems we employ here the number of elements is $N_{\boldsymbol{k}}^6 \approx 10^{13}$. *All* implicit integrators will therefore not be a viable option for FRG calculations.

Algorithm 1 Explicit adaptive Euler

```
1: $V = V_0$, $\Lambda = \Lambda_0$
2: while $\max(V) < V_{\max}$ and $\Lambda > 0$ do
3:     $dV = \mathrm{RHS}(V_n, \Lambda)$
4:     $V_{n+1} = V_n + \Delta_\Lambda dV$
5:     $\Lambda_{n+1} = \Lambda_n - \Delta_\Lambda$
6:     $\Delta_\Lambda = f(\Lambda, V_{\max})$
7: end while
```

2.2.4 Methods based on the non-linear Magnus series

Another class of one-step methods specifically suited for homogeneous, non-linear, non-autonomous ODEs is based on the Magnus form of the exact solution of (7),

$$V(\Lambda) = \exp\left(\Omega(\Lambda, V_\infty)\right) V_\infty \tag{11}$$

where $V_\infty = V(\Lambda = \infty)$ and we now instead require a suitable approximation for Ω. Ω fulfills a non-linear differential equation

$$\frac{d\Omega}{d\Lambda} = \mathrm{dexp}_\Omega^{-1}\left(A(\Lambda, e^\Omega V_\infty)\right), \quad \Omega(\infty) = 0 \tag{12}$$

where the inverse of the dexp_Ω operator is defined by the series

$$\mathrm{dexp}_\Omega^{-1}(C) = \sum_{k=0}^{\infty} \frac{B_k}{k!}\,\mathrm{ad}_\Omega^k C, \tag{13}$$

with the iterated right-nested commutator $\mathrm{ad}_X^{k+1}(Y) = [X, \mathrm{ad}_X^k(Y)]$, B_k are the well-known Bernoulli numbers, and the matrix A is defined by the relation $\mathrm{RHS}(\Lambda, V) = A(\Lambda, V)V$. The first-order approximation can be obtained by truncating Eq. (13) as

$$\Omega^{[1]}(\Lambda, \Lambda_\infty) = \int_{\Lambda_\infty}^{\Lambda} A(s, V_\infty)ds. \tag{14}$$

Of course the presence of iterated commutators makes higher order expressions unwieldy and hence we restrict ourselves to the first, Eq. (14), and second order approximations

$$\Omega^{[2]}(\Lambda, \Lambda_\infty) = \int_{\Lambda_\infty}^{\Lambda} A(s, e^{\Omega^{[1]}}V_\infty)ds. \tag{15}$$

The general higher order approximation can be found in the literature [46]. An iterative time-stepping procedure is obtained by applying the approximate time evolution operator $\exp(\Omega^{[m]})$ for small steps Δ_Λ such that

$$y_{n+1} = \exp\left(\Omega^{[m]}\left(\Lambda_n + \Delta_\Lambda, \Lambda_n\right)\right) y_n. \tag{16}$$

The integrals in Eqs. (14) and (15) are approximated by order-consistent low-order expressions, such that e.g., at first order

$$\Omega^{[1]}(\Lambda + \Delta_\Lambda, \Lambda) = \int_\Lambda^{\Lambda+\Delta_\Lambda} A(s, V(\Lambda))ds \tag{17}$$

$$\approx \Delta_\Lambda A(\Lambda, V(\Lambda)). \tag{18}$$

Adaptivity is easily included using time slices of differing times $\Delta_{\Lambda,n}$ determined by the same strategy as for the adaptive Euler.

2.2.5 Multistep/Adams methods

In contrast to one-step methods, multistep methods utilize previous evaluations of the right-hand side, $f_{n+k} = \mathrm{RHS}(V_{n+k}, \Lambda_{n+k})$ to approximate it with a Lagrange polynomial. The general form of these linear multistep methods can be stated as

$$\sum_{j=0}^{k} \alpha_j V_{n+j} = h \sum_{j=0}^{k} \beta_j \mathrm{RHS}(\Lambda_{n+j}, V_{n+j}). \tag{19}$$

If $\beta_k = 0$ the method is explicit and does not require the evaluation of RHS at unknown points. Commonly this class is termed Adams-Bashforth (AB) technique. Otherwise the method is implicit. If the interpolation polynomial is only augmented with the single yet unknown point $\mathrm{RHS}(\Lambda_{n+k}, V_{n+k})$ the method is termed Adams-Bashforth-Moulton method(ABM). In this case the implicit nature can be treated by employing a predictor-corrector scheme where the prediction $\hat{V}$ is obtained by an explicit Adams method of one order lower than the implicit method, RHS is evaluated and the new value V_{n+k} is obtained. Multistep methods need startup values, but these are obtained with explicit Runge–Kutta or similar methods. Adaptive step size integrators can be obtained by utilizing more sophisticated interpolation techniques.

2.3 Measures of quality

Differentiation of the quality of integrators will be based on the following aspects:

1. Error compared to a converged high-fidelity Euler calculation
2. Minimal number of RHS-evaluations
3. RAM requirements

2.3.1 Error compared to high-fidelity Euler

All integrators must converge to the same result at infinite numbers of steps performed, or equally at negligible integration error for adaptive procedures. To obtain this result in the most controlled fashion we converge an adaptive Euler integrator by successively shrinking the maximum allowed step-width. We find for all simulated parameter points, that a maximum step width of $\Delta_{\max} = 0.001\Lambda$ (cf. Sect. 2.2.2) leads to converged results, implying $N_{\mathrm{RHS}} = 5700 - 8000$ for a single simulation, depending on the precise divergence scale. Note that this number is an order of magnitude higher than for all other discussed methods. The measure of error employed in the comparison is the normed sum of differences in the resulting vertex tensor:

$$\frac{\sum_i |V(i) - V_{\mathrm{Euler}}(i)|}{\sum_i |V_{\mathrm{Euler}}(i)|}. \tag{20}$$

To avoid numerical discrepancies in the vicinity of the flow divergence, we instead compare the vertices slightly above the critical scale Λ_{crit} at $\Lambda = 0.17$. This scale is sufficiently low for the integrator to have performed a significant number of integration steps but sufficiently high to avoid the critical scale. Note that this number is arbitrarily chosen to lie within this region.

Because this measure does not include an analysis of the error size which is tolerable for qualitatively correct FRG results (for the given approximation level), we have additionally in stage 2 Sect. 3.2 included a full phase scan to ensure the results are consistent with expected behavior. We eliminate all integrators that are inconsistent.

2.3.2 Minimal number of RHS-evaluation

As we are aiming to optimize the calculation time requirements, the most important measure of quality for any integration routine must be the number of RHS-evaluations it requires until divergence. This is the only expensive part of the calculation and is thus a trivial, but platform independent estimator for the runtime. While this is no longer true for implicit integrators where the inversion of the Jacobian would be most expensive, these are disallowed by their RAM requirements and thus can be neglected here.

2.3.3 Memory requirement

For most FRG calculations the bottleneck is not only the calculation time but also the memory consumption [44,47]. As higher-order methods may require the saving of intermediate results we want to track the number of concurrently allocated vertex-sized objects as a measure of RAM usage. Once again for implicit methods this may not be the most accurate measure due to the creation of high-dimensional Jacobian matrices, but we disregard them entirely. The impact on memory usage by other objects is of second order when compared to the vertices in scaling: the vertices scale as $N_{\boldsymbol{k}}^3$ while the loop-derivatives scale as $N_{\boldsymbol{k}}^2$ and the dispersion cache scales as $N_{\boldsymbol{k}} N_{\boldsymbol{k}_f}$. The measure chosen is the peak memory usage during the calculations. This will be highly proportional to the total number of vertex objects though slight lower-order effects are to be expected.

3 Results

We have ensured that our implementation falls within the FRG equivalence class of the Hubbard model reported on in Ref. [44]. This is a grouping of three distinct FRG implementations which all reproduce numerically equivalent results for a narrow range of tests. Reproduction of all the published tests, most notably equality of all elements of the resulting vertex, confirms that this code is correct for the chosen approximations. Note that while one of the codes in that publication is written by one of us, we utilize for this benchmark an independent implementation written in the Julia programming language.

All simulations were performed using the following parameters: $t = 1, U = 3t, t' \in [0.0, ..., -0.5]t$, $\mu = 4t'$. We use a momentum meshing of $n_{\boldsymbol{k}} = 16 \times 16$ with a refined meshing of $n_{\text{fine}} = 9 \times 9$. The FRG flow is started at $\Lambda_\infty = 50$ and tracked until the maximum element of the vertex exceeds $V_{\max} = 50t$, where we terminate the integration and analyze the last vertex $V_{\Lambda_{\text{crit}}}$ with respect to its ordering tendencies. While this resolution is insufficient for physical simulations at low scales, and we acknowledge that the stability of all integrators increases for higher resolutions, the fact that most integrators yield correct phase predictions at the end of flow is testament to a sufficient resolution for this analysis. These predictions are shown in Fig. 3 where the similarities are easily apparent. We also already want to note that the limits of the chosen resolution are visible here in the difference to the established literature result. This gap could be lessened by increasing the resolution.

For error controlled adaptive step-size integrators, we allow for an absolute local error of 10^{-6}, a relative local error of 10^{-3} and set a lower bound of 10^{-5} on the absolute step size to prevent excessive runtimes close to divergences of the flow.

We have split the investigation into a two staged elimination process where we iterate over consecutively more involved tests to find the best set of integrators and neglect the remainder.

3.1 Stage 1

The first approach to determine the feasibility of the integrators is to analyze the square lattice Hubbard model at $t' = 0$, $\mu = 0$ where we expect antiferromagnetic ordering.

The results of stage 1 are displayed in Fig. 2. We now select the subset of integrators which yield a speedup (or are similar) in number of evaluations when compared to the adaptive Euler integration scheme while maintaining higher or similar accuracy. Other reasons for discarding, are that we run out of memory, lack of features, or plainly wrong results. A detailed listing of all considered integrators can be found in Appendix A, where we also give references to their details in literature. This reduces our list of integrators to consider for the continuation to be:

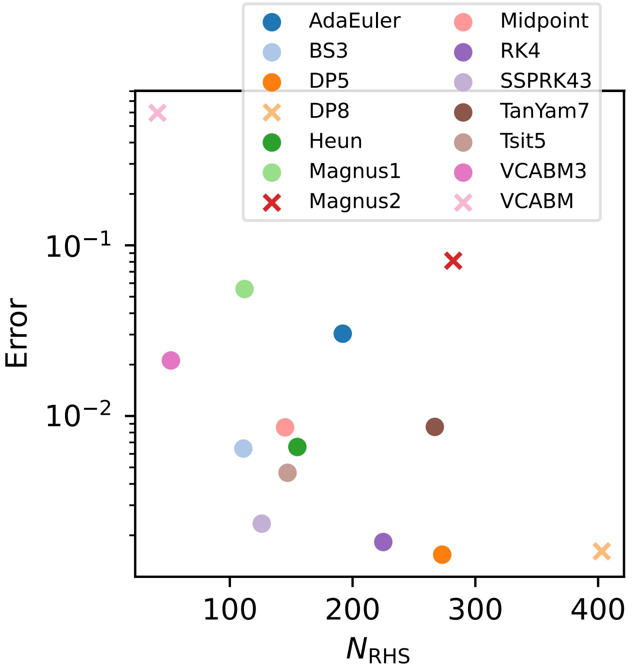

Fig. 2 Integrator Comparisons Quick analysis of the quality of integrators. For this measurement we chose the single point at $t' = 0$, $\mu = 0$ in the $t - t'$ square-lattice Hubbard model. At this point we expect an antiferromagnetic divergence, every method incapable of producing this is omitted. We show the number of steps required by the integrators for the prediction of the divergence as well as the error when compared to high-fidelity Euler calculations. The best integrators lie near the origin of the coordinate system, low number of steps and negligible error. We in the following will consider only integrators marked with circles, the set of which we will refer to as stage 2 integrators. The crossed ones were eliminated from further consideration as they were far from optimal. A full list of considered integrators as well as the reasons some are not shown can be found in Appendix A

- Adaptive Euler
- Bogacki-Shampine 3/2 (BS3)
- Dormand-Prince's 5/4 Runge–Kutta (DP5)
- Second order Heun's
- Second order Midpoint
- canonical Runge–Kutta 4 (RK4)
- Strong-stability preserving Runge–Kutta (SSPRK43)
- Tanaka-Yamashita 7 Runge–Kutta (TanYam7)
- Tsitouras 5/4 Runge–Kutta (Tsit5)
- Third order Adams-Moulton, BS3 for starting values (VCABM3)

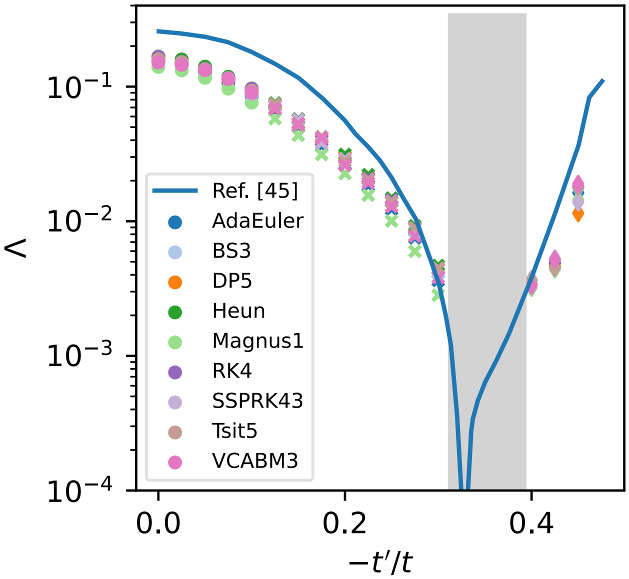

Fig. 3 Hubbard Model Calculations of the reference Hubbard model at van Hove filling with $t' \in [0.0, -0.4t]$. The comparative data is taken from [48], on which we have superimposed the subset of stage 2 integrators. We have removed all integrators who were unable to produce the correct phase at points. Note that we in this analysis ignore the region around the SC-FM phase transition as it is notoriously unstable under low scales [32,44]. Dots represent antiferromagnetism, crosses superconductivity and diamonds ferromagnetism. Note that the scale difference to literature results is due only to the chosen resolution

We have thus significantly reduced the number of integrators to consider in the next section. While we acknowledge that some of these might have been tweaked into compliance by optimizing parameters we insist on the out-of-the-box setup of the integrators being at least sufficient (if maybe not optimal). Particularly noteworthy is here the behavior of AB methods in contrast to the ABM methods. The AB methods sampled their startup values of V very close to the initial Λ_∞ where V is very smooth and structureless. From this structurelessness huge steps for the rest of the integration domain were inferred that led to wrong results. In the ABM methods this use of excessively huge step sizes was intrinsically prevented by the corrector step and hence did not require any tweaking of method parameters.

3.2 Stage 2

To check the reduced set of integrators for physical consistency we now evaluate the $t - t'$ square lattice Hubbard model at van Hove filling by scanning the second neighbor hopping t' (adapting the chemical potential $\mu = 4t'$ accordingly to remain at v.H. filling) as previously calculated in Refs. [1,32]. We perform these calculations with each of the second stage integrators to obtain a more qualified understanding of their accuracy and cost. The choice to evaluate along the van Hove singularity was made to ease the costs of the simulations. At the singularity the high density of states raises the critical scale of the transition, lowering the minimum required resolution to find a phase other than metal. We retain however quantitative differences in the critical scale compared to literature results as can be seen in Fig. 3.

To determine accuracy we in a first step check that the phase transitions occur in a controlled manner in all integrators and all points are properly found to diverge. In this we however make an exception near the SC-FM phase transitions, where the low integration resolutions used here *will* lead to uncontrolled behavior. By this process we eliminate the Midpoint and TanYam7 integration methods, which did not properly reproduce the leading instability for some points in the phase diagram. The remaining results and integrators are shown in Fig. 3. We can here also see the good agreement of qualitative results from the different integrators, an expected but satisfying result.

To evaluate the performance we sum the number of RHS evaluation for each of the 20 runs in the interval $t' \in [0, ..., -0.5]$ into a total number N_{RHS}. This is a more accurate measure of performance than before as the phase diagram contains parameter regions, where early divergences are expected as well as points, at which low RG scales have to be reached. This is supplemented by an improved error estimate: as before we calculate the effective vertex at the low scale of $\Lambda = 0.17$ and compare it to the results obtained by a converged high-fidelity Euler, but we now consider 3 additional points, one in each phase at $t' = 0.0, -0.05, -0.2, -0.45$. As an accuracy measure we consider the mean of the relative error for these four parameter combinations. The results of this analysis are presented in Fig. 4. We find, that depending on the exact requirements on the integrator, different algorithms should be employed. For purely optimizing the runtime, i.e. the number of RHS evaluations, the multistep method VCABM3 clearly is the method of choice, with a mean error comparable to the adaptive Euler, but only needing less than a quarter of RHS evaluations. On the other extreme, we find DP5 to be the most accurate, but requiring 40% more runtime. A competitive alternative is RK4 with a slightly larger relative deviation from the reference data, but at a computational cost comparable to the Adaptive Euler. As a good compromise between numerical complexity and accuracy, we find a cluster of three methods: BS3, Tsit5 and, most notably, SSPRK43 all have an order of magnitude lower relative error, but at roughly two third of the computational cost of the Adaptive Euler.

In Fig. 5 we additionally show the measured peak memory requirement of our implementation relative to the Adaptive Euler. This figure is highly implementation dependent, but should nevertheless serve as a rough guide to the overhead incurred by the different solvers. Most clearly, this figure shows, that the extreme numerical efficiency of VCABM3 is achieved trading runtime for memory consumption, with a fivefold requirement compared to a simple Euler implemen-

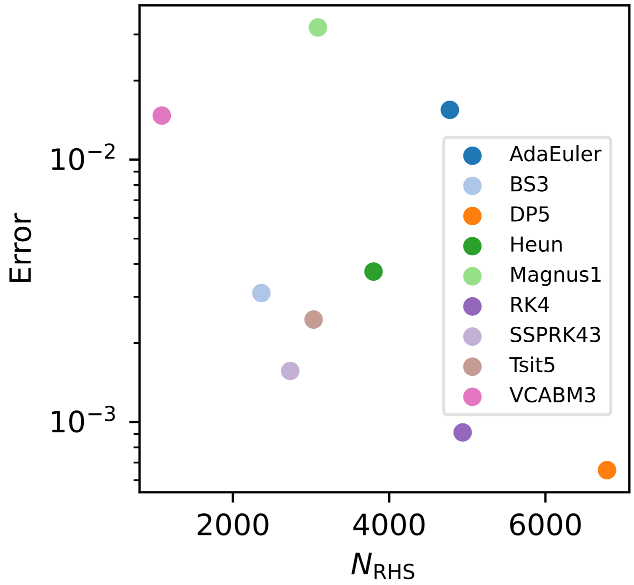

Fig. 4 Stage 2 Integrator Comparisons We compare the integrators which passed the primary analysis in a second iteration using an extended testing scheme. This extended scheme consists of a full $t - t'$ Hubbard model phase scan. The N_{RHS} here represents the total number of steps for the evaluation of the 20 points in the phase diagram. Similarly, the error represents the mean of the errors at the four evaluation points $t' = 0, -0.05, -0.2, -0.4$. Once again proximity to the origin is the quality measure in the graphic

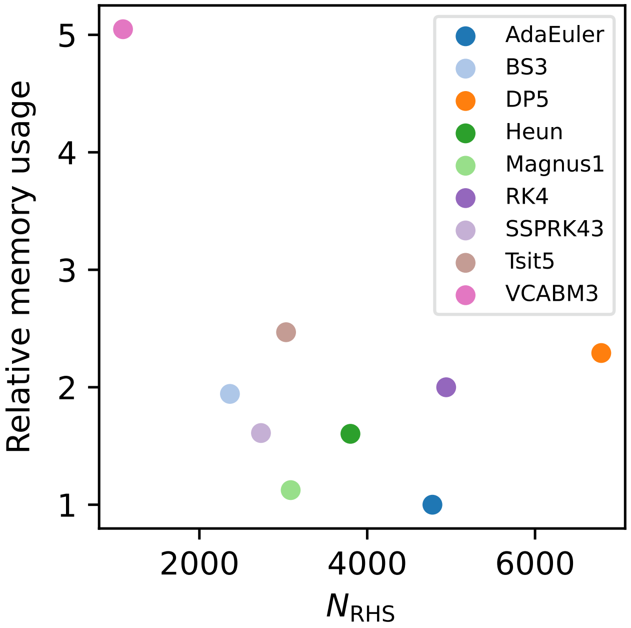

Fig. 5 Stage 2 Memory Comparison Supplementing the analysis presented in Fig. 4 we here provide the measurements of the memory usage during the integration procedure. We have denoted the memory usage in terms of the most efficient schemes, the adaptive Euler. Note that we have just measured the total memory usage, partial might thus correspond to objects of smaller size. Nonetheless, the measurement will reflect the ram needed for the integrations very well

tation. The remaining integrators considered all fall in the range of 1.5 to 2.5 the memory requirement. Out of the three general algorithms, SSPRK43 clearly has the lowest memory requirements, which combined with its good accuracy and numerical efficiency makes it the best suited ODE integrator in our benchmark, followed closely by BS3.

4 Conclusions

We have benchmarked a multitude of ODE integration algorithms both from the `DifferentialEquations.jl` library and own implementations against a reference Adaptive Euler using the square-lattice Hubbard model as a test case. We have identified, that the right choice of integrator can significantly reduce the numerical cost for integrating the runtime while at the same time producing more accurate results. We identify the multistep algorithm VCABM3 to be the most efficient one with acceptable numerical errors, while highest accuracy can be achieved by the numerically expensive DP5 algorithm, or alternatively the slightly less accurate RK4, which however has about the same numerical cost as the Adaptive Euler.

As the best compromise between accuracy and numerical speedup, we identify Tsit5, BS3 and SSPRK43, with the latter slightly outclassing its rivals. This fact at first glance is surprising, as this specific integrator is designed to handle dissipative differential equations stemming from a discretization of hyperbolic partial differential equations. However, as the FRG equations are believed to flow towards a fixed point for any given phase [49], we speculate them to exhibit dissipation in the mathematical sense, which is the property SSPRK43 is design to utilize. This means that solutions of the RG flow corresponding to different initial conditions in the same phase become more similar as they approach $\Lambda = 0$, as the dominant physics will be the same for all of these. A mathematically thorough analysis of the FRG equations regarding this property, however, is beyond the scope of this paper.

As we only have analyzed a subset of the plethora of available ODE integrators in this paper, we cannot claim to have found the “best” integrator for FRG system, but we ensured to represent a spectrum of the common choices.

We also have focused on the simplest physically interesting model, the square-lattice Hubbard model. While we believe, that the scaling of computational cost can be extrapolated to other problems, more complicated problems posed with more intricate Fermi surfaces may require denser integration steps, the best algorithms may therefore differ.

Furthermore, when using other approximations of momentum space FRG, for example self energy inclusions [50] or multiloop [9,16,51,52], the RHS equations are subject to structural changes. The inclusion of self energies might benefit from the use of operator splitting methods, such as the generalized Strang or Leapfrog splitting [53], where self energy and vertex are evaluated at alternating half-step intervals. The combination of operator splittings together with the Magnus series could also enhance its competitiveness when certain parts of the equations can be treated analytically exact.

The merit of our results is still valid, the best integrators as found here will be good candidates for the other applications.

Another path of algorithmic progress can be made by taking into account the intricacies of the FRG equations. Since their integration starts at infinity we propose to investigate transformations of this semi-infinite domain as in Refs. [20,54,55].

Furthermore, the integration of the FRG equations contains an inherent singularity corresponding to the physical ordering tendency. Implicit methods [56] are well suited for such problems, but were out of the scope of our investigation due to memory requirements. It could be a worthwhile direction of future investigation. More specialized multistep integrators that utilize rational interpolants [56,57] instead of Newton and Lagrange polynomials for the definition of their integration rule could be beneficial in dealing with the singular behavior. But of course their effectiveness rests on methods of obtaining the action of the Jacobian of RHS or suitable approximations to it.

Acknowledgements. J.B. thanks GRF (German Research Foundation) for support through RTG 1995 within the Priority Program SPP 244 “2DMP”. F.G. and T.M. thank the GRF for funding through the SFB 1170 “Tocotronics” under the grant numbers Z03 and B04, respectively. Furthermore, we want to thank Jonas Hauck, Dominik Kiese, Lennart Klebl, Janik Potten, Stephan Rachel and Tilman Schwemmer for helpful discussions.

Author contributions

J.B. and T.M. were leading the development of the code with F.G. taking charge of the simulation effort. Analysis was conducted on a joint basis, the manuscript was a joint effort by the authors. Overall no distinction can be drawn in level of contribution between the authors.

Funding Open Access funding enabled and organized by Projekt DEAL.

Data Availability Statement This manuscript has associated data in a data repository. [Authors’ comment: The FRG implementation used in this paper is available at https://doi.org/10.5281/zenodo.6391339. The numerical data shown is available from the corresponding author upon reasonable request.]

Appendix A Full list of considered integrators

The full list of single-step and multi-step integrators for differential equations considered in this work can be found in Tables 1 and 2, respectively, together with the corresponding literature reference and the reason we have discarded them, if applicable.

Table 1 Full set of considered single-step integrators. The disregarded integrators have the reason given for disregarding them

Name	Reference	Reason if discarded
Adaptive Euler	[45]	
Adaptive Magnus1	[46]	
Adaptive Magnus2	[46]	(Error Stage 1)
BS3	[58]	
ARKODE	[59]	(Complex arithmetic)
DP5	[60]	
DP8	[45]	(Runtime Stage 1)
EPIRK4s3A	[42]	(Not adaptive)
exp4	[42]	(Not adaptive)
exprb32	[42]	
Feagin10	[61]	(Wrong Stage 1)
FRK65	[42]	(Very excessive runtime)
Heun	[45]	
HochOst4	[42]	(Not adaptive)
implicit Euler	[56]	(Memory)
implicit Midpoint	[56]	(Memory)
KenCarp4	[42]	(Memory)
Kvaerno5	[42]	(Memory)
LawsonEuler	[42]	(Memory)
Midpoint	[45]	(Wrong Hubbard)
NorsettEuler	[42]	(Not adaptive)
RadauIIa3, RadauIIa5	[42]	(Complex arithmetic)
RK4	[45]	
RKO65	[42]	(Not adaptive)

Table 1 continued

Name	Reference	Reason if discarded
ROCK2, ROCK4	[42]	(Excessive runtime)
Rodas4p2	[42]	(Memory)
Ros3p	[42]	(Memory)
Rosenbrock23	[42]	(Memory)
SDIRK2	[42]	(Memory)
SSPRK22, SSPRK33, SSPRK83	[42]	(Not adaptive)
SSPRK43	[62]	
SSPRK932	[42]	(Type errors, excessive runtime)
TanYam7	[63]	(Wrong Hubbard)
Trapezoid	[56]	(Memory)
Tsit5	[64]	
veldd4, velds4	[42]	(Memory)

Table 2 Full set of considered multi-step integrators. The disregarded integrators have the reason given for disregarding them

Name	Reference	Reason if discarded
AB3, AB5, ABM32, ABM54	[45]	(Not adaptive)
QBD, QNDF, QBDF1, QNDF1	[42]	(Memory)
TRBDF2	[42]	(Memory)
VCAB3	[45]	(Error Stage 1)
VCAB5	[45]	(Error Stage 1)
VCABM3	[45]	
VCABM5	[45]	(Error Stage 1)

References

1. M. Salmhofer, C. Honerkamp, Fermionic renormalization group flows: technique and theory. Progress Theoretical Phys. **105**(1), 1–35 (2001)
2. W. Metzner, M. Salmhofer, C. Honerkamp, V. Meden, K. Schönhammer, Functional renormalization group approach to correlated fermion systems. Rev. Mod. Phys. **84**, 299–352 (2012). https://doi.org/10.1103/RevModPhys.84.299
3. C. Platt, W. Hanke, R. Thomale, Functional renormalization group for multi-orbital fermi surface instabilities. Adv. Phys. **62**(4–6), 453–562 (2013). https://doi.org/10.1080/00018732.2013.862020. arXiv: 1310.6191
4. J. Reuther, P. Wölfle, J_1-J_2 frustrated two-dimensional Heisenberg model: random phase approximation and functional renormalization group. Phys. Rev. B **81**, 144410 (2010)
5. J. Reuther, R. Thomale, Cluster functional renormalization group. Phys. Rev. B **89**(2), 024412 (2014)
6. Y. Iqbal, R. Thomale, F. Parisen Toldin, S. Rachel, J. Reuther, Functional renormalization group for three-dimensional quantum magnetism. Phys. Rev. B **94**, 140408 (2016)
7. F.L. Buessen, M. Hering, J. Reuther, S. Trebst, Quantum spin liquids in frustrated spin-1 diamond antiferromagnets. Phys. Rev. Lett. **120**, 057201 (2018)
8. Y. Iqbal, T. Müller, P. Ghosh, M.J.P. Gingras, H.O. Jeschke, S. Rachel, J. Reuther, R. Thomale, Quantum and classical phases of the pyrochlore Heisenberg model with competing interactions. Phys. Rev. X **9**, 011005 (2019)
9. A. Tagliavini, C. Hille, F. Kugler, S. Andergassen, A. Toschi, C. Honerkamp, Multiloop functional renormalization group for the two-dimensional hubbard model: loop convergence of the response functions. SciPost Phys. **6**(1), 009 (2019)
10. F.L. Buessen, The SpinParser software for pseudofermion functional renormalization group calculations on quantum magnets (2021) arXiv:2109.13317
11. D. Kiese, T. Müller, Y. Iqbal, R. Thomale, S. Trebst, Multiloop functional renormalization group approach to quantum spin systems. Phys. Rev. Res. **4**, 023185 (2022). https://doi.org/10.1103/PhysRevResearch.4.023185
12. J. Thoenniss, M. Ritter, F. Kugler, J. von Delft, M. Punk, Multiloop pseudofermion functional renormalization for quantum spin systems: Application to the spin-1/2 kagome heisenberg model. arXiv preprint arXiv:2011.01268 (2020)
13. P. Chalupa-Gantner, F.B. Kugler, C. Hille, J. von Delft, S. Andergassen, A. Toschi, Fulfillment of sum rules and ward identities in the multiloop functional renormalization group solution of the anderson impurity model. Phys. Rev. Res. **4**, 023050 (2022). https://doi.org/10.1103/PhysRevResearch.4.023050
14. P.M. Bonetti, A. Toschi, C. Hille, S. Andergassen, D. Vilardi, Single-boson exchange representation of the functional renormalization group for strongly interacting many-electron systems. Phys. Rev. Res. **4**, 013034 (2022). https://doi.org/10.1103/PhysRevResearch.4.013034
15. C. Hille, F.B. Kugler, C.J. Eckhardt, Y.-Y. He, A. Kauch, C. Honerkamp, A. Toschi, S. Andergassen, Quantitative functional renormalization group description of the two-dimensional hubbard model. Phys. Rev. Res. **2**(3), 033372 (2020)
16. F.B. Kugler, J. von Delft, Multiloop functional renormalization group that sums up all parquet diagrams. Phys. Rev. Lett. **120**(5), 057403 (2018)
17. F.B. Kugler, J. Von Delft, Fermi-edge singularity and the functional renormalization group. J. Phys. **30**(19), 195501 (2018)
18. L. Markhof, B. Sbierski, V. Meden, C. Karrasch, Detecting phases in one-dimensional many-fermion systems with the functional renormalization group. Phys. Rev. B **97**(23) (2018). https://doi.org/10.1103/physrevb.97.235126
19. L. Weidinger, F. Bauer, J. von Delft, Functional renormalization group approach for inhomogeneous one-dimensional fermi systems with finite-ranged interactions. Phys. Rev. B **95**(3) (2017). https://doi.org/10.1103/physrevb.95.035122
20. F. Bauer, J. Heyder, J. von Delft, Functional renormalization group approach for inhomogeneous interacting fermi systems. Phys. Rev. B **89**, 045128 (2014). https://doi.org/10.1103/PhysRevB.89.045128

21. J.B. Hauck, C. Honerkamp, S. Achilles, D.M. Kennes, Electronic instabilities in penrose quasicrystals: Competition, coexistence, and collaboration of order. Phys. Rev. Res. **3**(2) (2021). https://doi.org/10.1103/physrevresearch.3.023180
22. C. Honerkamp, M. Salmhofer, Temperature-flow renormalization group and the competition between superconductivity and ferromagnetism. Phys. Rev. B **64**, 184516 (2001). https://doi.org/10.1103/PhysRevB.64.184516
23. C. Honerkamp, Electron-doping versus hole-doping in the 2d t-t' hubbard model. Eur. Phys. J. B **21**(1), 81–91 (2001). https://doi.org/10.1007/PL00011117
24. C. Honerkamp, M. Salmhofer, N. Furukawa, T.M. Rice, Breakdown of the landau-fermi liquid in two dimensions due to umklapp scattering. Phys. Rev. B **63**, 035109 (2001). https://doi.org/10.1103/PhysRevB.63.035109
25. C. Honerkamp, M. Salmhofer, Flow of the quasiparticle weight in the n-patch renormalization group scheme. Phys. Rev. B **67**, 174504 (2003). https://doi.org/10.1103/PhysRevB.67.174504
26. C. Honerkamp, D. Rohe, S. Andergassen, T. Enss, Interaction flow method for many-fermion systems. Phys. Rev. B **70**(23), 235115 (2004). https://doi.org/10.1103/PhysRevB.70.235115
27. C. Honerkamp, M. Salmhofer, Ferromagnetism and triplet superconductivity in the two-dimensional Hubbard model. Physica C 408-410, 302–304 (2004). https://doi.org/10.1016/j.physc.2004.02.089. arXiv: cond-mat/0307541
28. M. Salmhofer, C. Honerkamp, W. Metzner, O. Lauscher, Renormalization group flows into phases with broken symmetry. Progress Theoretical Phys. 112(6), 943–970 (2004). https://doi.org/10.1143/PTP.112.943. arXiv: cond-mat/0409725
29. C. Husemann, M. Salmhofer, Efficient parametrization of the vertex function, ω scheme, and the t, t' hubbard model at van hove filling. Phys. Rev. B **79**(19), 195125 (2009)
30. S. Uebelacker, C. Honerkamp, Self-energy feedback and frequency-dependent interactions in the functional renormalization group flow for the two-dimensional hubbard model. Phys. Rev. B **86**(23) (2012). https://doi.org/10.1103/physrevb.86.235140
31. C. Husemann, K.-U. Giering, M. Salmhofer, Frequency dependent vertex functions of the (t, t')-Hubbard model at weak coupling. Phys. Rev. B **85**(7), 075121 (2012). https://doi.org/10.1103/PhysRevB.85.075121 arXiv: 1111.6802
32. J. Lichtenstein, Functional renormalization group studies on competing orders in the square lattice. Dissertation, RWTH Aachen University, Aachen (2018). https://doi.org/10.18154/RWTH-2018-225781. published on the publications server of the RWTH Aachen University; Dissertation, RWTH Aachen University, 2018. https://publications.rwth-aachen.de/record/728603
33. C. Hille, D. Rohe, C. Honerkamp, S. Andergassen, Pseudogap opening in the two-dimensional Hubbard model: A functional renormalization group analysis. Phys. Rev. Res. 2(3), 033068 (2020). https://doi.org/10.1103/PhysRevResearch.2.033068. arXiv: 2003.01447
34. T. Schäfer, N. Wentzell, F. Šimkovic, Y.-Y. He, C. Hille, M. Klett, C.J. Eckhardt, B. Arzhang, V. Harkov, F.m.c.-M. Le Régent, A. Kirsch, Y. Wang, A.J. Kim, E. Kozik, E.A. Stepanov, A. Kauch, S. Andergassen, P. Hansmann, D. Rohe, Y.M. Vilk, J.P.F. LeBlanc, S. Zhang, A.-M.S. Tremblay, M. Ferrero, O. Parcollet, A. Georges, Tracking the footprints of spin fluctuations: A multimethod, multimessenger study of the two-dimensional hubbard model. Phys. Rev. X 11, 011058 (2021). https://doi.org/10.1103/PhysRevX.11.011058
35. D. Vilardi, C. Taranto, W. Metzner, Nonseparable frequency dependence of the two-particle vertex in interacting fermion systems. Phys. Rev. B **96**, 235110 (2017). https://doi.org/10.1103/PhysRevB.96.235110
36. T. Reckling, C. Honerkamp, Approximating the frequency dependence of the effective interaction in the functional renormalization group for many-fermion systems. Phys. Rev. B **98**(8), 085114 (2018). https://doi.org/10.1103/PhysRevB.98.085114 arXiv: 1803.08431
37. R. Thomale, C. Platt, J. Hu, C. Honerkamp, B.A. Bernevig, Functional renormalization-group study of the doping dependence of pairing symmetry in the iron pnictide superconductors. Phys. Rev. B **80**(18), 180505 (2009)
38. M.L. Kiesel, C. Platt, W. Hanke, D.A. Abanin, R. Thomale, Competing many-body instabilities and unconventional superconductivity in graphene. Phys. Rev. B **86**(2) (2012). https://doi.org/10.1103/physrevb.86.020507
39. W.-S. Wang, Y.-Y. Xiang, Q.-H. Wang, F. Wang, F. Yang, D.-H. Lee, Functional renormalization group and variational monte carlo studies of the electronic instabilities in graphene near $\frac{1}{4}$ doping. Phys. Rev. B **85**, 035414 (2012). https://doi.org/10.1103/PhysRevB.85.035414
40. J. Lichtenstein, S.A. Maier, C. Honerkamp, C. Platt, R. Thomale, O.K. Andersen, L. Boeri, Functional renormalization group study of an eight-band model for the iron arsenides. Phys. Rev. B **89**(21) (2014). https://doi.org/10.1103/physrevb.89.214514
41. L. Klebl, Q. Xu, A. Fischer, L. Xian, M. Claassen, A. Rubio, D.M. Kennes, Moiré engineering of spin-orbit coupling in twisted platinum diselenide. Electronic Structure (2022)
42. C. Rackauckas, Q. Nie, Differentialequations.jl–a performant and feature-rich ecosystem for solving differential equations in julia. J. Open Res. Softw. **5**(1) (2017)
43. Boost Community: Boost Odeint. boost.org
44. Jacob Beyer, Jonas B. Hauck, Lennart Klebl, Reference results for the momentum space functional renormalization group. Eur. Phys. J. B **95**(4), 65 (2022). https://doi.org/10.1140/epjb/s10051-022-00323-y
45. E. Hairer, S.P. Nørsett, G. Wanner, Solving Ordinary Differential Equations I, Nonstiff Problems, Corr. 3rd Printing. Springer, Berlin (2008). https://doi.org/10.1007/978-3-540-78862-1
46. F. Casas, A. Iserles, Explicit magnus expansions for nonlinear equations. J. Phys. A **39**(19), 5445–5461 (2006). https://doi.org/10.1088/0305-4470/39/19/s07
47. J. Lichtenstein, D. Sánchez de la Peña, D. Rohe, E.D. Napoli, C. Honerkamp, S.A. Maier, High-performance functional Renormalization Group calculations for interacting fermions. Comput. Phys. Commun. **213**, 100–110 (2017). https://doi.org/10.1016/j.cpc.2016.12.013
48. C. Honerkamp, M. Salmhofer, Magnetic and superconducting instabilities of the hubbard model at the van hove filling. Phys. Rev. Lett. **87**(18) (2001). https://doi.org/10.1103/physrevlett.87.187004

49. W. Metzner, M. Salmhofer, C. Honerkamp, V. Meden, K. Schönhammer, Functional renormalization group approach to correlated fermion systems. Rev. Mod. Phys. **84**, 299–352 (2012). https://doi.org/10.1103/RevModPhys.84.299
50. K.-U. Giering, M. Salmhofer, Self-energy flows in the two-dimensional repulsive hubbard model. Phys. Rev. B **86**(24), 245122 (2012)
51. F.B. Kugler, J. von Delft, Multiloop functional renormalization group for general models. Phys. Rev. B **97**(3), 035162 (2018)
52. F.B. Kugler, J. von Delft, Derivation of exact flow equations from the self-consistent parquet relations. New J. Phys. **20**(12), 123029 (2018)
53. R.I. McLachlan, On the numerical integration of ordinary differential equations by symmetric composition methods. SIAM J. Sci. Comput. **16**(1), 151–168 (1995). https://doi.org/10.1137/0916010
54. W. Qin, C. Huang, T. Wolf, N. Wei, I. Blinov, A.H. MacDonald, Functional Renormalization Group Study of Superconductivity in Rhombohedral Trilayer Graphene. arXiv (2022). https://doi.org/10.48550/ARXIV.2203.09083. https://arxiv.org/abs/2203.09083
55. Jonas B. Hauck, Dante M. Kennes, TU^{2}FRG: a scalable approach for truncated unity functional renormalization group in generic fermionic models. Eur. Phys. J. B **95**(3), 60 (2022). https://doi.org/10.1140/epjb/s10051-022-00316-x
56. G. Wanner, E. Hairer, Solving Ordinary Differential Equations II vol. 375. Springer, Berlin (1996)
57. A. Karimov, D. Butusov, V. Andreev , E.G. Nepomuceno, Rational approximation method for stiff initial value problems. Mathematics **9**(24) (2021). https://doi.org/10.3390/math9243185
58. P. Bogacki, L.F. Shampine, A 3 (2) pair of runge-kutta formulas. Appl. Math. Lett. **2**(4), 321–325 (1989)
59. A.C. Hindmarsh, P.N. Brown, K.E. Grant, S.L. Lee, R. Serban, D.E. Shumaker, C.S. Woodward, SUNDIALS: Suite of nonlinear and differential/algebraic equation solvers. ACM Trans. Math. Softw. (TOMS) **31**(3), 363–396 (2005)
60. J.R. Dormand, P.J. Prince, A family of embedded runge-kutta formulae. J. Comput. Appl. Math. **6**(1), 19–26 (1980)
61. T. Feagin, High-order explicit runge-kutta methods using m-symmetry. Neural Parallel Sci. Comput. **20**(3–4), 437–458 (2012)
62. J.F.B.M. Kraaijevanger, Contractivity of runge-kutta methods. BIT **31**(3), 482–528 (1991). https://doi.org/10.1007/BF01933264
63. M. Tanaka, E. Kasahara, S. Muramatsu, S. Yamashita, On a solution of the order conditions for the nine-stage seventh order explicit runge-kutta method. Inform. Process. Soc. Jpn. **33**(12), 1506–1511 (1992)
64. C. Tsitouras, Runge-kutta pairs of order 5 (4) satisfying only the first column simplifying assumption. Comput. Math. Appl. **62**(2), 770–775 (2011)