Drudge/Gristmill: A Full-Stack Solution for Quantum Many-Body Theory Development

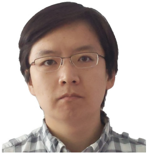
Guo P. Chen

Scuseria Group, Department of Chemistry, Rice University

June 5, 2023

Acknowledgements

Drudge (https://github.com/tschijnmo/drudge)
Gristmill (https://github.com/tschijnmo/gristmill)

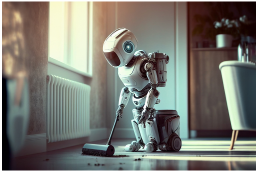


Dr. Jinmo Zhao

- Dr. Gaurav Harsha
- Dr. Thomas Henderson
- Prof. Gustavo Scuseria

Why Build New Tools?

- `Drudge`
  - Noncommutative algebras
  - Structured tensors
  - Symbolic summation
- `Gristmill`
  - Working equation optimization
  - Code generation

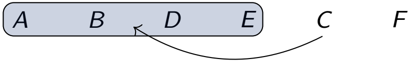


General Commutation Algebra

$$XY - e^{i\varphi(X,Y)}YX = S(X,Y)$$

- CAR algebra (fermions), CCR algebra (bosons), Lie algebras
- Normal ordering of generators

$$A \prec B \prec C \prec D \prec E \prec F$$



- `GenQuadDrudge` $\succ$ `GenQuadLatticeDrudge` $\succ$ `SU2LatticeDrudge`
  - Pairing $su(2)$ generated by $(P^\dagger, P, N)$

$$P_p^\dagger = c_p^\dagger c_{\bar{p}}^\dagger, \quad N_p = c_p^\dagger c_p + c_{\bar{p}}^\dagger c_{\bar{p}}$$

  - Antisymmetrized geminal power (AGP) or PBCS

T. M. Henderson, G. E. Scuseria, *J. Chem. Phys.* **153**, 084111 (2020)
A. Khamoshi, G. P. Chen, T. M. Henderson, G. E. Scuseria, *J. Chem. Phys* **154**, 074113 (2021)
R. Dutta, G. P. Chen, T. M. Henderson, G. E. Scuseria, *J. Chem. Phys* **154**, 114112 (2021)

Wickian Algebra

$$XY - e^{i\varphi(X,Y)}YX = \sigma(X,Y)$$

where

$$\varphi(X,Y) = -\varphi(Y,X), \quad \sigma(X,Y) = -\sigma(Y,X)\, e^{-i\varphi(Y,X)}$$

- CAR algebra (fermions), CCR algebra (bosons)
- Wick's theorem for normal ordering
- Matrix elements between mean-field states

$$\begin{aligned}\langle\Phi_0|c_p^\dagger c_q^\dagger c_s c_r|\Phi_1\rangle &= \langle\Phi_0|c_p^\dagger c_q^\dagger c_s c_r|\Phi_1\rangle + \langle\Phi_0|c_p^\dagger c_q^\dagger c_s c_r|\Phi_1\rangle + \langle\Phi_0|c_p^\dagger c_q^\dagger c_s c_r|\Phi_1\rangle \\ &= \langle\Phi_0|\Phi_1\rangle\left(\rho_{rp}\rho_{sq} - \rho_{rq}\rho_{sp} + \chi_{pq}\kappa_{rs}\right)\end{aligned}$$

- `WickDrudge` $\succ$ `FockDrudge` $\succ$ `GenMBDrudge` $\succ$ `PartHoleDrudge` $\succ$ `SpinOneHalfPartHoleDrudge`

Canonicalization

$$u_{pqrs} = -u_{pqsr} = -u_{qprs} = u_{qpsr}$$

$$\frac{1}{4}\sum_{rs} u_{prqs}\rho_{sr}c_p^\dagger c_q + \frac{1}{4}\sum_{rs} u_{sprq}\rho_{rs}c_p^\dagger c_q - \frac{1}{4}\sum_{rs} u_{psrq}\rho_{rs}c_p^\dagger c_q - \frac{1}{4}\sum_{rs} u_{rpqs}\rho_{sr}c_p^\dagger c_q = \sum_{rs} u_{prqs}\rho_{sr}c_p^\dagger c_q$$

```
dr.set_symm(u, Perm([1, 0, 3, 2], IDENT), Perm([0, 1, 3, 2], NEG))
```

Jinmo Zhao, *"Symbolic solution for computational quantum many-body theory development,"* Ph.D. thesis, Rice University (2018)

Example: Coupled Cluster Singles and Doubles

$$|\Psi\rangle = e^{T} |\Phi\rangle = e^{T_1 + T_2} |\Phi\rangle$$

$$T_1 = \sum_{ia} t^{(1)}_{ai} c^\dagger_a c_i, \quad T_2 = \frac{1}{4} \sum_{ijab} t^{(2)}_{abij} c^\dagger_a c^\dagger_b c_j c_i$$

$$\bar{H} = e^{-T} H e^{T}$$

$$E_c = \langle \Phi | \bar{H} | \Phi \rangle$$

$$r^{(1)}_{ai} = \langle \Phi | c^\dagger_i c_a \bar{H} | \Phi \rangle = 0$$

$$r^{(2)}_{abij} = \langle \Phi | c^\dagger_i c^\dagger_j c_b c_a \bar{H} | \Phi \rangle = 0$$

Example: Symmetry-Projected Hartree–Fock–Bogoliubov

- Hartree–Fock–Bogoliubov (HFB)

$$|\Phi\rangle = \prod_p \beta_p |-\rangle$$

$$\beta_p^\dagger = \sum_q \left(U_{qp} c_q^\dagger + V_{qp} c_q \right)$$

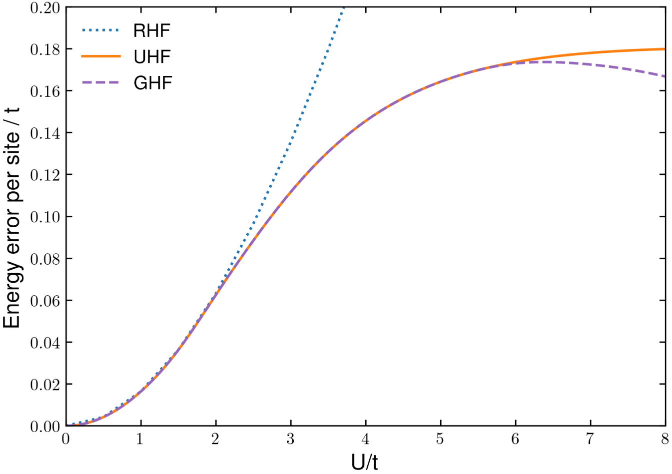
- Symmetry projection

$$|\Psi\rangle = \mathcal{P}\,|\Phi\rangle = \int \mathrm{d}\mu(g)\,|\Phi(g)\rangle$$

$$E = \frac{\langle\Psi|\mathcal{H}|\Psi\rangle}{\langle\Psi|\Psi\rangle} = \frac{\langle\Phi|\mathcal{P}^\dagger\mathcal{H}\mathcal{P}|\Phi\rangle}{\langle\Phi|\mathcal{P}^\dagger\mathcal{P}|\Phi\rangle} = \frac{\langle\Phi|\mathcal{H}\mathcal{P}|\Phi\rangle}{\langle\Phi|\mathcal{P}|\Phi\rangle}$$

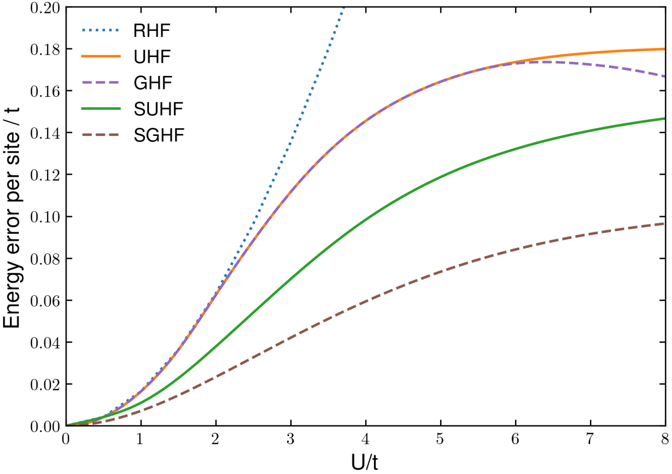
$$H(g) = \langle\Phi|\mathcal{H}|\Phi(g)\rangle$$

Doped Hubbard Chain



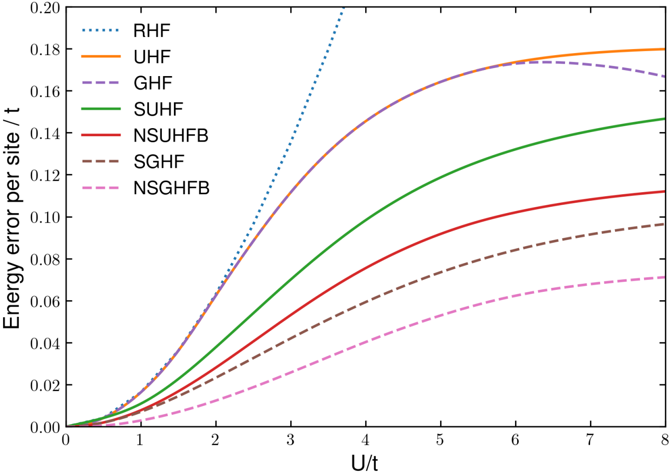
10-site Hubbard chain with 8 electrons

Application to Doped Hubbard Chain



10-site Hubbard chain with 8 electrons

Application to Doped Hubbard Chain



10-site Hubbard chain with 8 electrons

Matrix Elements

- Nonorthogonal Wick's theorem

$$\begin{aligned}\langle \Phi_0 | c_p^\dagger c_q^\dagger c_s c_r | \Phi_1 \rangle &= \langle \Phi_0 | c_p^\dagger c_q^\dagger c_s c_r | \Phi_1 \rangle + \langle \Phi_0 | c_p^\dagger c_q^\dagger c_s c_r | \Phi_1 \rangle + \langle \Phi_0 | c_p^\dagger c_q^\dagger c_s c_r | \Phi_1 \rangle \\ &= \langle \Phi_0 | \Phi_1 \rangle \left(\rho_{rp} \rho_{sq} - \rho_{rq} \rho_{sp} + \chi_{pq} \kappa_{rs} \right)\end{aligned}$$

- Contractions

$$\rho_{pq} = c_q^\dagger c_p, \quad \kappa_{pq} = c_q c_p, \quad \chi_{pq} = c_p^\dagger c_q^\dagger$$

```python
class ProjectedHFBDrudge(GenMBDrudge):
    ...

    @property
    def hfb_contractor(self):
        '''

    def eval_hfb_transition(self, tensor: Tensor):
        return self.eval_vev(tensor, self.hfb_contractor)

    ...
```

Matrix Elements of the Hamiltonian

$$H = \sum_{pq} h_{pq} c_p^\dagger c_q + \frac{1}{4} \sum_{pqrs} u_{pqrs} c_p^\dagger c_q^\dagger c_s c_r$$

```
dr = ProjectedHFBDrudge(ctx)

tm = dr.ham.eval_hfb_transition()
tm.display()
```

$$\sum_{p\in L} \sum_{q\in L} \sum_{r\in L} \sum_{s\in L} \frac{1}{2} \rho_{p,r} \rho_{q,s} u_{r,s,p,q} + \sum_{p\in L} \sum_{q\in L} \sum_{r\in L} \sum_{s\in L} \frac{1}{4} \chi_{p,q} \kappa_{r,s} u_{p,q,r,s} + \sum_{p\in L} \sum_{q\in L} h_{p,q} \rho_{p,q}$$

Numerical Difficulties

- Nonorthogonal Wick's theorem

$$\langle\Phi_0|c_p^\dagger c_q^\dagger c_s c_r|\Phi_1\rangle = \langle\Phi_0|\Phi_1\rangle\left(\rho_{rp}\rho_{sq} - \rho_{rq}\rho_{sp} + \chi_{pq}\kappa_{rs}\right)$$

$$\rho_{pq} = \frac{\langle\Phi_0|c_q^\dagger c_p|\Phi_1\rangle}{\langle\Phi_0|\Phi_1\rangle}, \quad \kappa_{pq} = \frac{\langle\Phi_0|c_q c_p|\Phi_1\rangle}{\langle\Phi_0|\Phi_1\rangle}, \quad \chi_{pq} = \frac{\langle\Phi_0|c_p^\dagger c_q^\dagger|\Phi_1\rangle}{\langle\Phi_0|\Phi_1\rangle}$$

- Robust Wick's theorem

$$\langle\Phi_0|\Phi_1\rangle = \mathrm{pf}(\mathcal{S}) = \zeta\prod_r s_r$$

$$\rho_{pq} = \sum_r s_r^{-1}\tilde{\rho}_{pq}^r, \quad \kappa_{pq} = \sum_r s_r^{-1}\tilde{\kappa}_{pq}^r, \quad \chi_{pq} = \sum_r s_r^{-1}\tilde{\chi}_{pq}^r$$

$$\langle\Phi_0|c_p^\dagger c_q^\dagger c_{q'} c_{p'}|\Phi_1\rangle = \zeta\sum_{r_1 r_2}\lambda_{r_1 r_2}\left(\tilde{\rho}_{p'p}^{r_1}\tilde{\rho}_{q'q}^{r_2} - \tilde{\rho}_{p'q}^{r_1}\tilde{\rho}_{q'p}^{r_2} + \tilde{\chi}_{pq}^{r_1}\tilde{\kappa}_{p'q'}^{r_2}\right)$$

G. P. Chen, G. E. Scuseria, `arXiv:2304.13780` (accepted by *J. Chem. Phys.*, editor's pick)

Low-Scaling Robust Wick's Theorem

$$\rho = \rho^{S} + \rho^{R} = \sum_{r \in S} s_r^{-1} \tilde{\rho}^{r} + \rho^{R}$$

$$\begin{aligned}
\langle \Phi_0 | c_p^\dagger c_q^\dagger c_{q'} c_{p'} | \Phi_1 \rangle &= \langle \Phi_0 | \Phi_1 \rangle \left(\rho_{pq}^{01,R} \rho_{p'q'}^{01,R} + \rho_{pq}^{01,S} \rho_{p'q'}^{01,R} + \rho_{pq}^{01,R} \rho_{p'q'}^{01,S} + \rho_{pq}^{01,S} \rho_{p'q'}^{01,S} \right) \\
&= \lambda^{R} \left(\lambda^{S} \rho_{pq}^{01,R} \rho_{p'q'}^{01,R} + \sum_{r_1 \in S} \lambda_{r_1}^{S} \tilde{\rho}_{pq}^{01,r_1} \rho_{p'q'}^{01,R} \right. \\
&\qquad \left. + \sum_{r_1 \in S} \lambda_{r_1}^{S} \rho_{pq}^{01,R} \tilde{\rho}_{p'q'}^{01,r_1} + \sum_{r_1, r_2 \in S} \lambda_{r_1 r_2}^{S} \tilde{\rho}_{pq}^{01,r_1} \tilde{\rho}_{p'q'}^{01,r_2} \right)
\end{aligned}$$

G. P. Chen, G. E. Scuseria, `arXiv:2304.13780` (accepted by *J. Chem. Phys.*, editor's pick)

Low-Scaling Robust Wick's Theorem

$$\rho = \rho^S + \rho^R = \sum_{r \in S} s_r^{-1} \tilde{\rho}^r + \rho^R$$

```python
rs_partition = [
    dr.define(rho[p, q], rho_R[p, q] + rho_S[p, q]),
    dr.define(kappa[p, q], kappa_R[p, q] + kappa_S[p, q]),
    dr.define(chi[p, q], chi_R[p, q] + chi_S[p, q]),
]
tm = tm.subst_all(rs_partition)
```

$$\sum_{p\in L}\sum_{q\in L}\sum_{r\in L}\sum_{s\in L}\frac{1}{2}\rho^{(R)}_{p,r}\rho^{(R)}_{q,s}u_{r,s,p,q} + \sum_{p\in L}\sum_{q\in L}\sum_{r\in L}\sum_{s\in L}\frac{1}{2}\rho^{(R)}_{p,r}\rho^{(S)}_{q,s}u_{r,s,p,q} + \sum_{p\in L}\sum_{q\in L}\sum_{r\in L}\sum_{s\in L}\frac{1}{2}\rho^{(R)}_{q,s}\rho^{(S)}_{p,r}u_{r,s,p,q}$$
$$+ \sum_{p\in L}\sum_{q\in L}\sum_{r\in L}\sum_{s\in L}\frac{1}{2}\rho^{(S)}_{p,r}\rho^{(S)}_{q,s}u_{r,s,p,q} + \sum_{p\in L}\sum_{q\in L}\sum_{r\in L}\sum_{s\in L}\frac{1}{4}\chi^{(R)}_{p,q}\kappa^{(R)}_{r,s}u_{p,q,r,s} + \sum_{p\in L}\sum_{q\in L}\sum_{r\in L}\sum_{s\in L}\frac{1}{4}\chi^{(S)}_{p,q}\kappa^{(R)}_{r,s}u_{p,q,r,s}$$
$$+ \sum_{p\in L}\sum_{q\in L}\sum_{r\in L}\sum_{s\in L}\frac{1}{4}\chi^{(R)}_{p,q}\kappa^{(S)}_{r,s}u_{p,q,r,s} + \sum_{p\in L}\sum_{q\in L}\sum_{r\in L}\sum_{s\in L}\frac{1}{4}\chi^{(S)}_{p,q}\kappa^{(S)}_{r,s}u_{p,q,r,s} + \sum_{p\in L}\sum_{q\in L}h_{p,q}\rho^{(R)}_{p,q} + \sum_{p\in L}\sum_{q\in L}h_{p,q}\rho^{(S)}_{p,q}$$

Low-Scaling Robust Wick's Theorem

```python
def expand_singular_part_per_term(term: Term):
    assert len(term.vecs) == 0
    factors, coeff = term.amp_factors

    order = 0
    factors_new = []
    for factor in factors:
        if factor.base.name.endswith("_S"):
            indices_new = factor.indices + (s_dumms[order],)
            factors_new.append(factor.base[indices_new])
            order += 1
        else:
            factors_new.append(factor)

    if order == 0:
        amp = lambda0 * term.amp
        sums = term.sums
    else:
        dumms = s_dumms[:order]
        amp = coeff * lambda_bases[order-1][dumms] * Mul(*factors_new)
        sums = term.sums + tuple((dumm, s_range) for dumm in dumms)

    return Term(sums, amp, ())

tm = tm.map(expand_singular_part_per_term)
tm.display()
```

Low-Scaling Robust Wick's Theorem

$$
\begin{aligned}
&\sum_{p\in L}\sum_{q\in L}\sum_{r\in L}\sum_{s\in L}\frac{\lambda_0}{2}\rho^{(R)}_{p,r}\rho^{(R)}_{q,s}u_{r,s,p,q}+\sum_{p\in L}\sum_{q\in L}\sum_{r\in L}\sum_{s\in L}\sum_{r_1\in S}\frac{1}{2}\lambda^{(1)}_{r_1}\rho^{(R)}_{p,r}\rho^{(S)}_{q,s,r_1}u_{r,s,p,q}\\
&+\sum_{p\in L}\sum_{q\in L}\sum_{r\in L}\sum_{s\in L}\sum_{r_1\in S}\frac{1}{2}\lambda^{(1)}_{r_1}\rho^{(R)}_{q,s}\rho^{(S)}_{p,r,r_1}u_{r,s,p,q}+\sum_{p\in L}\sum_{q\in L}\sum_{r\in L}\sum_{s\in L}\sum_{r_1\in S}\sum_{r_2\in S}\frac{1}{2}\lambda^{(2)}_{r_1,r_2}\rho^{(S)}_{p,r,r_1}\rho^{(S)}_{q,s,r_2}u_{r,s,p,q}\\
&+\sum_{p\in L}\sum_{q\in L}\sum_{r\in L}\sum_{s\in L}\frac{\lambda_0}{4}\chi^{(R)}_{p,q}\kappa^{(R)}_{r,s}u_{p,q,r,s}+\sum_{p\in L}\sum_{q\in L}\sum_{r\in L}\sum_{s\in L}\sum_{r_1\in S}\frac{1}{4}\lambda^{(1)}_{r_1}\kappa^{(R)}_{r,s}\chi^{(S)}_{p,q,r_1}u_{p,q,r,s}\\
&+\sum_{p\in L}\sum_{q\in L}\sum_{r\in L}\sum_{s\in L}\sum_{r_1\in S}\frac{1}{4}\lambda^{(1)}_{r_1}\chi^{(R)}_{p,q}\kappa^{(S)}_{r,s,r_1}u_{p,q,r,s}+\sum_{p\in L}\sum_{q\in L}\sum_{r\in L}\sum_{s\in L}\sum_{r_1\in S}\sum_{r_2\in S}\frac{1}{4}\lambda^{(2)}_{r_1,r_2}\chi^{(S)}_{p,q,r_1}\kappa^{(S)}_{r,s,r_2}u_{p,q,r,s}\\
&+\sum_{p\in L}\sum_{q\in L}\lambda_0 h_{p,q}\rho^{(R)}_{p,q}+\sum_{p\in L}\sum_{q\in L}\sum_{r_1\in S}\lambda^{(1)}_{r_1}h_{p,q}\rho^{(S)}_{p,q,r_1}
\end{aligned}
$$

- FLOP count of `Gristmill` optimized code: $\mathcal{O}(4m^4 n_S)$
  - m: size of the one-particle basis
  - n_S: size of the (near-)zero $\{s_r\}$

Exploiting Sparsity

- Sparsity in h_{pq} and u_{pqrs}
- Hubbard Hamiltonian

$$H = -\sum_{pq}\sum_{\sigma\in\{\uparrow,\downarrow\}} t_{pq}\left(c^{\dagger}_{p\sigma}c_{q\sigma} + h.c.\right) + U\sum_{p} n_{p\uparrow}n_{p\downarrow}$$

$$\begin{aligned}
&-\sum_{p\in L}\sum_{q\in L}\lambda_0\rho^{(R00)}_{p,q}t_{p,q} - \sum_{p\in L}\sum_{q\in L}\sum_{r_1\in S}\lambda^{(1)}_{r_1}t_{p,q}\rho^{(S00)}_{p,q,r_1} - \sum_{p\in L}\sum_{q\in L}\lambda_0\rho^{(R11)}_{p,q}t_{p,q}\\
&-\sum_{p\in L}\sum_{q\in L}\sum_{r_1\in S}\lambda^{(1)}_{r_1}t_{p,q}\rho^{(S11)}_{p,q,r_1} + \sum_{p\in L}U\lambda_0\chi^{(R01)}_{p,p}\kappa^{(R01)}_{p,p} + \sum_{p\in L}\sum_{r_1\in S}U\lambda^{(1)}_{r_1}\kappa^{(R01)}_{p,p}\chi^{(S01)}_{p,p,r_1}\\
&+\sum_{p\in L}\sum_{r_1\in S}U\lambda^{(1)}_{r_1}\chi^{(R01)}_{p,p}\kappa^{(S01)}_{p,p,r_1} + \sum_{p\in L}\sum_{r_1\in S}\sum_{r_2\in S}U\lambda^{(2)}_{r_1,r_2}\chi^{(S01)}_{p,p,r_1}\kappa^{(S01)}_{p,p,r_2} + \sum_{p\in L}U\lambda_0\rho^{(R00)}_{p,p}\rho^{(R11)}_{p,p}\\
&+\sum_{p\in L}\sum_{r_1\in S}U\lambda^{(1)}_{r_1}\rho^{(R00)}_{p,p}\rho^{(S11)}_{p,p,r_1} + \sum_{p\in L}\sum_{r_1\in S}U\lambda^{(1)}_{r_1}\rho^{(R11)}_{p,p}\rho^{(S00)}_{p,p,r_1} + \sum_{p\in L}\sum_{r_1\in S}\sum_{r_2\in S}U\lambda^{(2)}_{r_1,r_2}\rho^{(S00)}_{p,p,r_1}\rho^{(S11)}_{p,p,r_2}\\
&-\sum_{p\in L}U\lambda_0\rho^{(01)}_{p,p}\rho^{(10)}_{p,p}
\end{aligned}$$

Matrix Elements of Higher-Body Operators

- Gradient w.r.t. Thouless rotation

$$\langle \Phi_0 | \beta_q^0 \, \beta_p^0 \, H | \Phi_1 \rangle$$

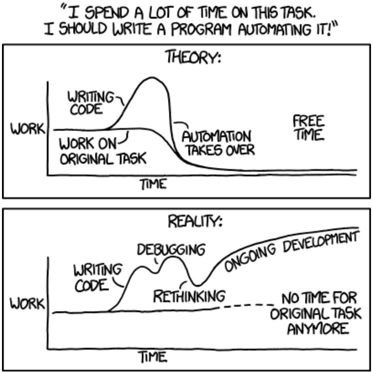
- Kernels in PHFB-based correlated methods
  - Projected coupled cluster

    Y. Qiu, T. M. Henderson, T. Duguet, G. E. Scuseria, *Phys. Rev. C* **99**, 044301 (2019)

Summary

Drudge (https://github.com/tschijnmo/drudge)
Gristmill (https://github.com/tschijnmo/gristmill)

- Automated derivation, optimization, and code generation
- Correctness, generality, and efficiency
- Interactive and reproducible
- Python ecosystem
- Parallel



xkcd.com/1319

J. Zhao, PhD thesis, Rice University (2018)

G. P. Chen, G. E. Scuseria, arXiv:2304.13780 (2023)