FIRE: Fast Inertial Relaxation Engine for Optimization on All Scales

P. Koskinen

E. Bitzek, F. Gähler, M. Moseler, and P. Gumbsch

Fraunhofer
Institut
Werksoffmechanik

izbs

Funding: Academy of Finland
Fraunhofer MAVO for Multiscale Materials Modeling

Optimization

Local optimization:
- structure optimization
- constrained optimization
- transition state (barrier) calculations
- stability analysis
- etc...

Global minimization
- often uses local minimization methods

Optimization

Local optimization:
- structure optimization
- constrained optimization
- transition state (barrier) calculations
- stability analysis
- etc...

Toolbox:

Steepest Descent (SD)
Conjugate Gradient (CG)
Molecular Dynamics (MD)
    ('quenching')
Quasi-Newton (QN)
    (BFGS, L-BFGS)
Truncated Newton (TN)
etc...

Choose your weapon

Toolbox:

Steepest Descent (SD)
Conjugate Gradient (CG)
Molecular Dynamics (MD)
('quenching')
Quasi-Newton (QN)
(BFGS, L-BFGS)
Truncated Newton (TN)
etc...

Choose your weapon

Toolbox:

- Steepest Descent (SD)
- Conjugate Gradient (CG)
- Molecular Dynamics (MD)
  ('quenching')
- Quasi-Newton (QN)
  (BFGS, L-BFGS)
- Truncated Newton (TN)
- etc...

Choose your weapon

- computational cost
  - function calls
  - computational overhead
- memory requirements $\sim N \leftrightarrow N^2$
- robustness
- easy to use? (parameters, implementation)
- convergence criteria (δE, $\mathbf{F}$, max($\mathbf{F}$), $\delta \mathbf{r}$,...)

Toolbox:

Steepest Descent (SD)
Conjugate Gradient (CG)
Molecular Dynamics (MD) ('quenching')
Quasi-Newton (QN) (BFGS, L-BFGS)
Truncated Newton (TN)
etc...

Choose your weapon

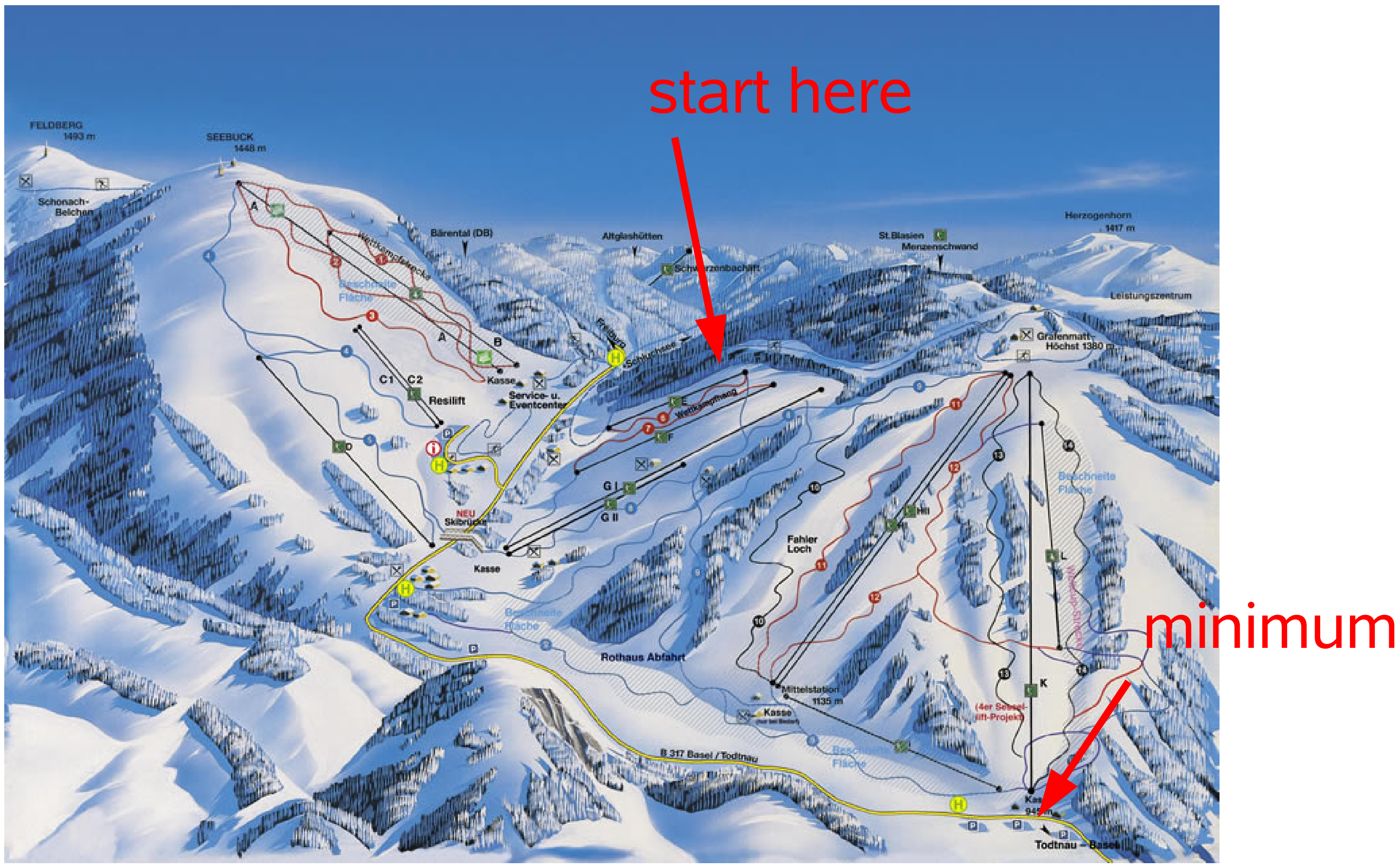
- computational cost
  - function calls
  - computational overhead
- memory requirements ~ N↔N^2
- robustness
- easy to use? (parameters, implementation)
- convergence cr
  (δE, **F**, max(**F**

Molecular Dynamics (MD) ('quenching')

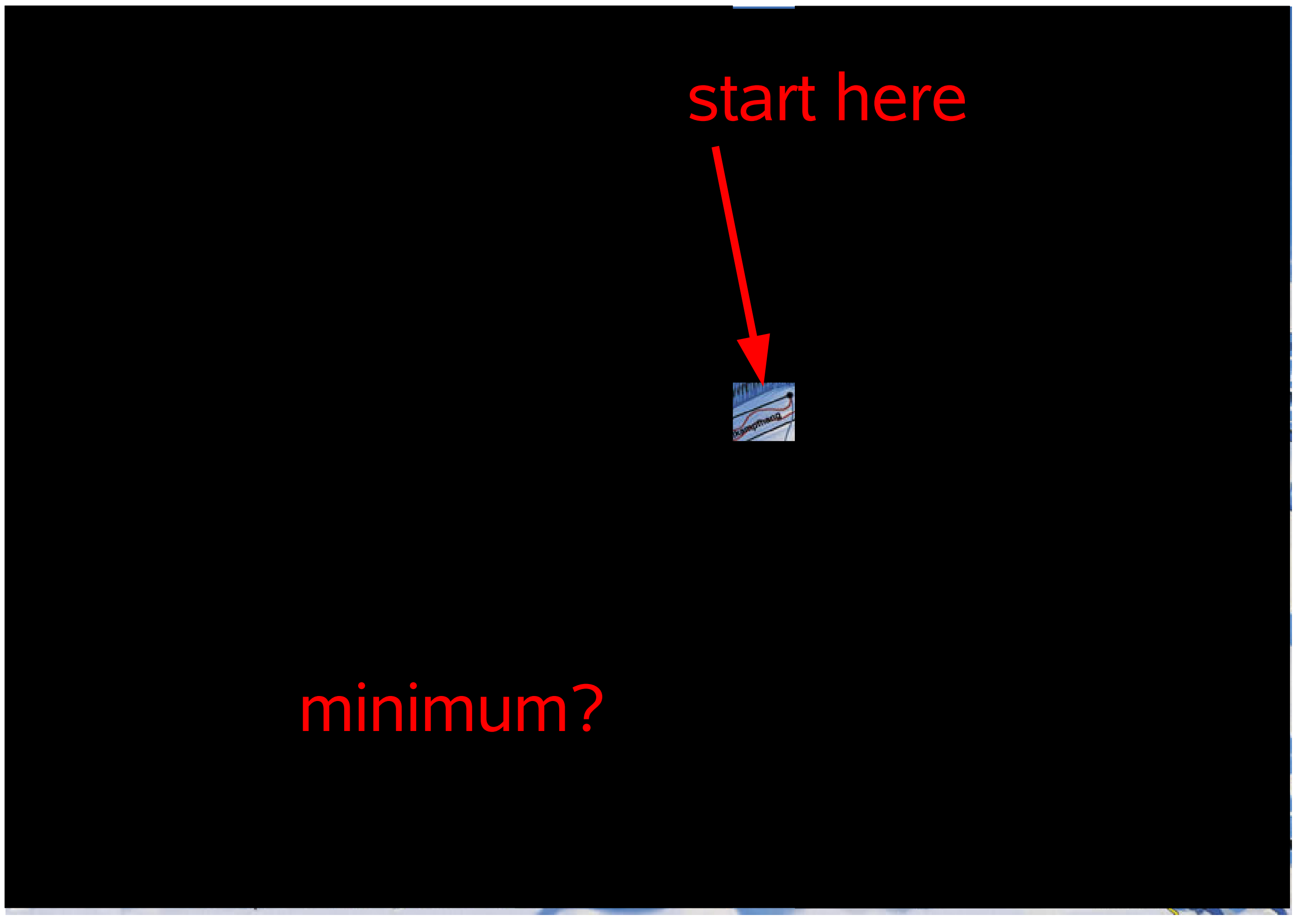
Toolbox:

Steepest Descent (SD)
Conjugate Gradient (CG)
Molecular Dynamics (MD) ('quenching')
Quasi-Newton (QN) (BFGS, L-BFGS)
Truncated Newton (TN)
etc…

Clever skier



Clever (blind) skier

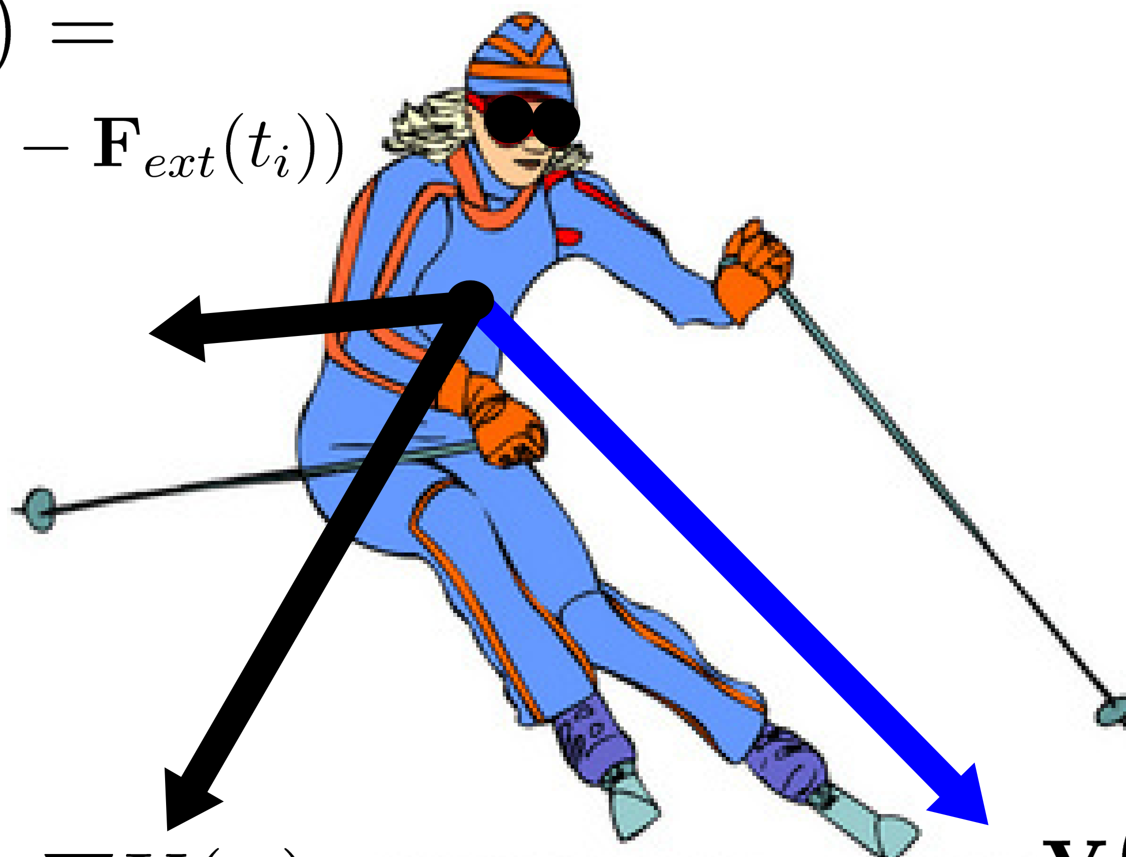


Snapshot at t=t$_i$

$$\mathbf{F}_{ext}(t_i) = -\nabla U(\mathbf{x}) \qquad \mathbf{v}(t_i)$$

Snapshot at $t=t_i$

$$\mathbf{F}_{skier}(t_i) =$$
$$-\gamma(t_i)|\mathbf{v}(t_i)|(\mathbf{v}(t_i) - \mathbf{F}_{ext}(t_i))$$



$$\mathbf{v}(t_i)$$

$$\mathbf{F}_{ext}(t_i) = -\nabla U(\mathbf{x})$$

Considerations

- if we go uphill ⇒ stop
- use MD with discrete Δt: optimize Δt in a stable manner:
  - right direction? ⇒increase Δt
  - where to go? ⇒decrease Δt
  - Δt should have max limit
  - no hasty decisions
- steer in the beginning, then let it go
  - too heavy steering ⇒ SD
  - let *inertia* decide the direction

↳ Fast Inertial Relaxation Engine (FIRE)

All dimensions should be comparable
vectors are 3N-dimensional

The algorithm

MD: calculate x, $\mathbf{F} = -\nabla E(\mathbf{x})$, and v using any common MD integrator; check for convergence

F1: calculate $P = \mathbf{F} \cdot \mathbf{v}$

F2: set $\mathbf{v} \rightarrow (1-\alpha) \cdot \mathbf{v} + \alpha \cdot \mathbf{F}/|\mathbf{F}| \cdot |\mathbf{v}|$

F3: if $P > 0$ and the number of steps since P was negative is larger than N_{min}, increase the time step $\Delta t \rightarrow \min(\Delta t \cdot f_{inc}, \Delta t_{max})$ and decrease $\alpha \rightarrow \alpha \cdot f_{\alpha}$

F4: if $P \leq 0$, decrease time step $\Delta t \rightarrow \Delta t \cdot f_{dec}$, freeze the system $v \rightarrow 0$ and set α back to α_{start}

F5: return to MD

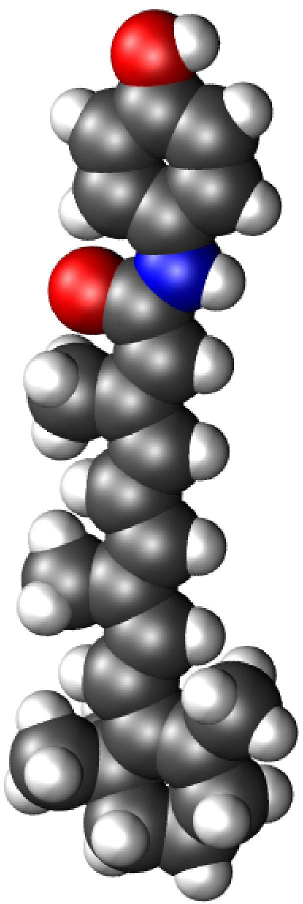
E. Bizek, P. Koskinen, F. Gähler, M. Moseler, P. Gumbch, *PRL* (to appear)

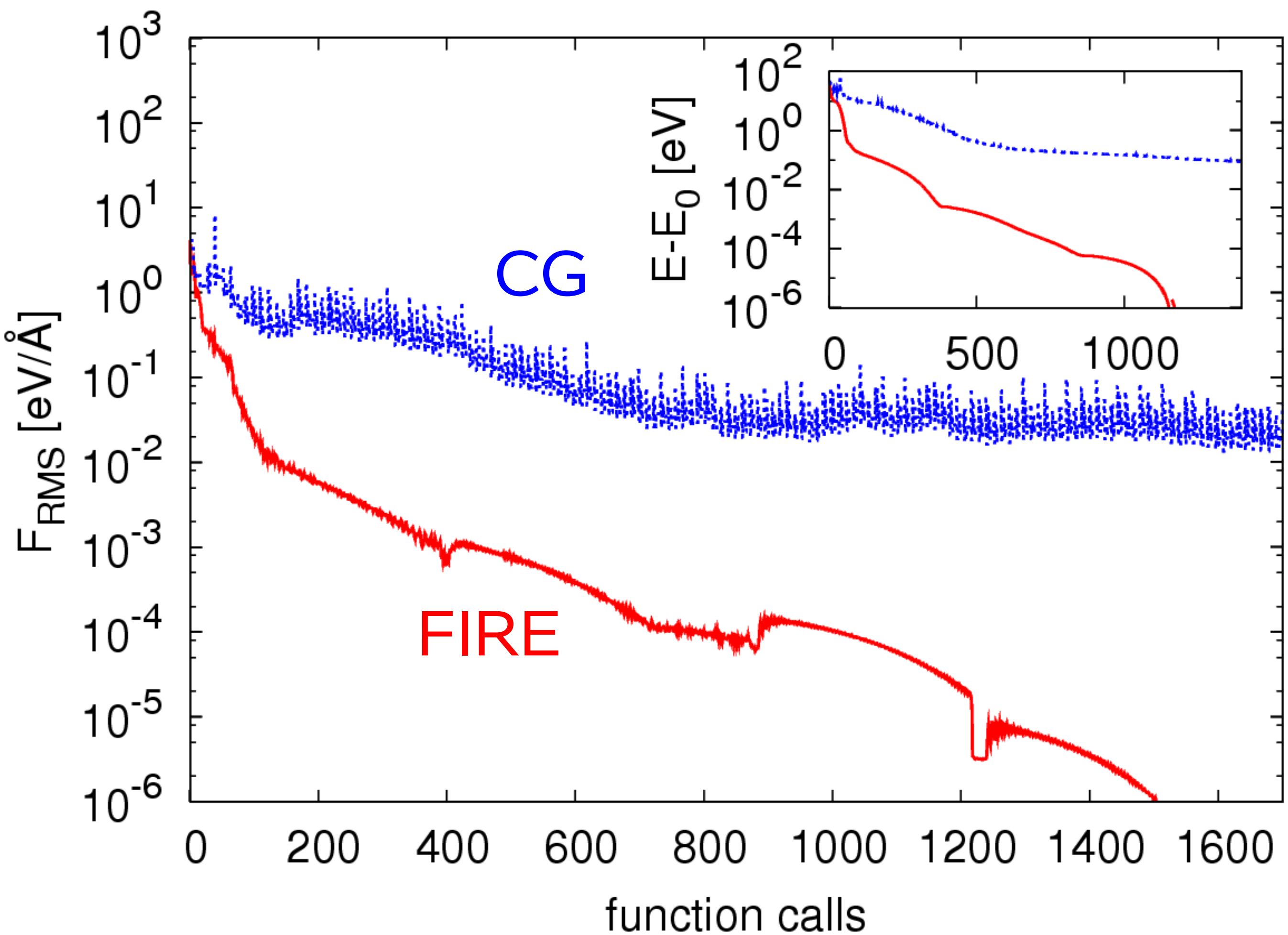
Sample code (!)

```
subroutine FIRE(it,dt,n,v[n],F[n],done)
  if( max(F) < crit ) done

  P = V*F
  V = (1-a)*V + a*F*norm(V)/norm(F)

  if( P<0 )
    V   = 0
    cut = it
    dt  = dt * f_dec
    a   = a_start
  else if( P>=0 and it-cut>N_min )
    dt  = min( dt*f_inc, dt_max )
    a   = a * f_dec
```

Results

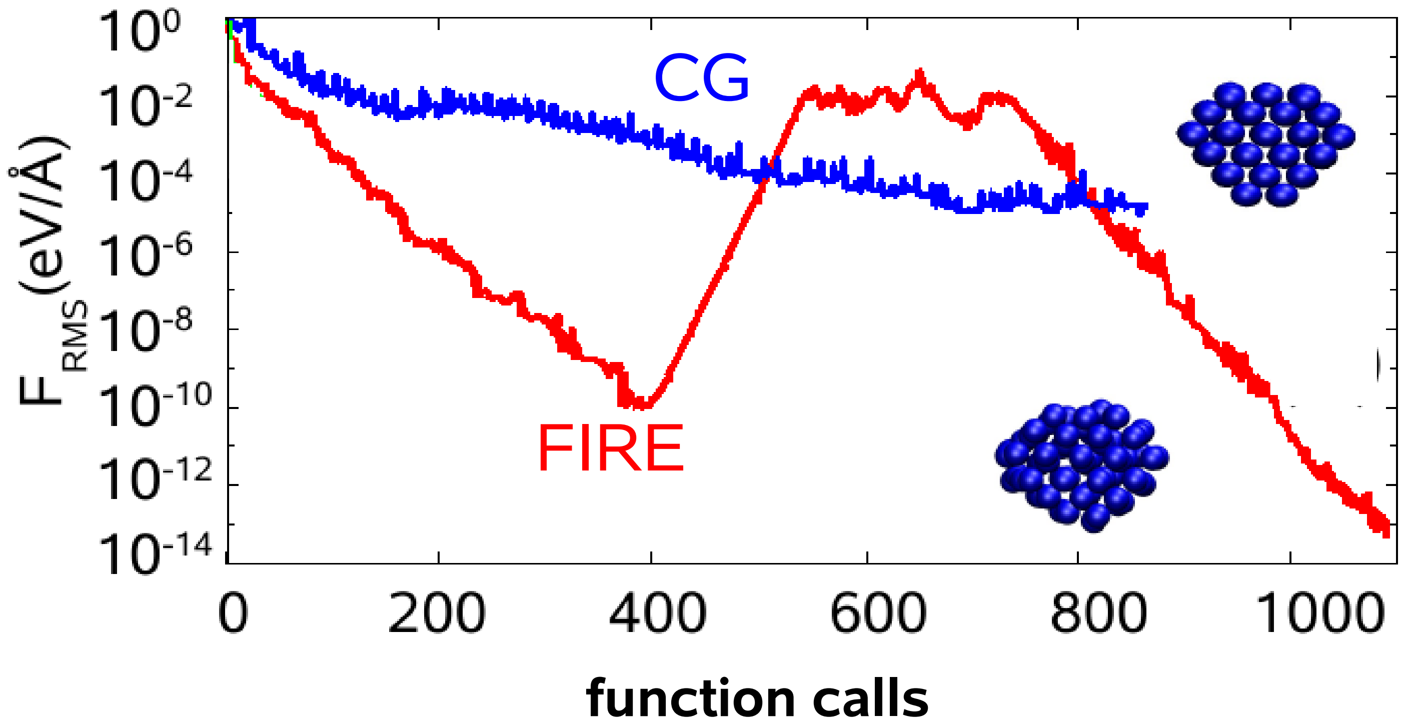
Fenretinide (N=62)





Results

Small convergence criteria: Na_{71}^-



Results

Convergence criteria: F_{RMS} or $F_{max} < 10^{-3}$ eV/Å   (10^{-6} eV/Å)

system	N_{atoms}	FIRE	CG
AlNiCo	3360	136 (639)	661 (2131)
crack in Ni	4815	61 (207)	174 (764)
hot Cu plate	16200	299 (585)	545 (1767)
vacancy in Cu	107998	43 (132)	58 (329)
vacancy in Cu	1492991	43 (118)	59 (358)

Summary

FIRE competes with sophisticated methods!

Summary

FIRE competes with sophisticated methods!

Furthermore:

- robust relaxation for all N
- memory usage small
- small computational overhead
- gradient-based: stability analysis
- gradient based: small convergence criteria
- non-harmonic energy landscapes
- stable against errors in E(**x**) and **F**(**x**)
- constrained minimization easy
- intuitive, easy adaption to new problems
- application to non-atomistic problems

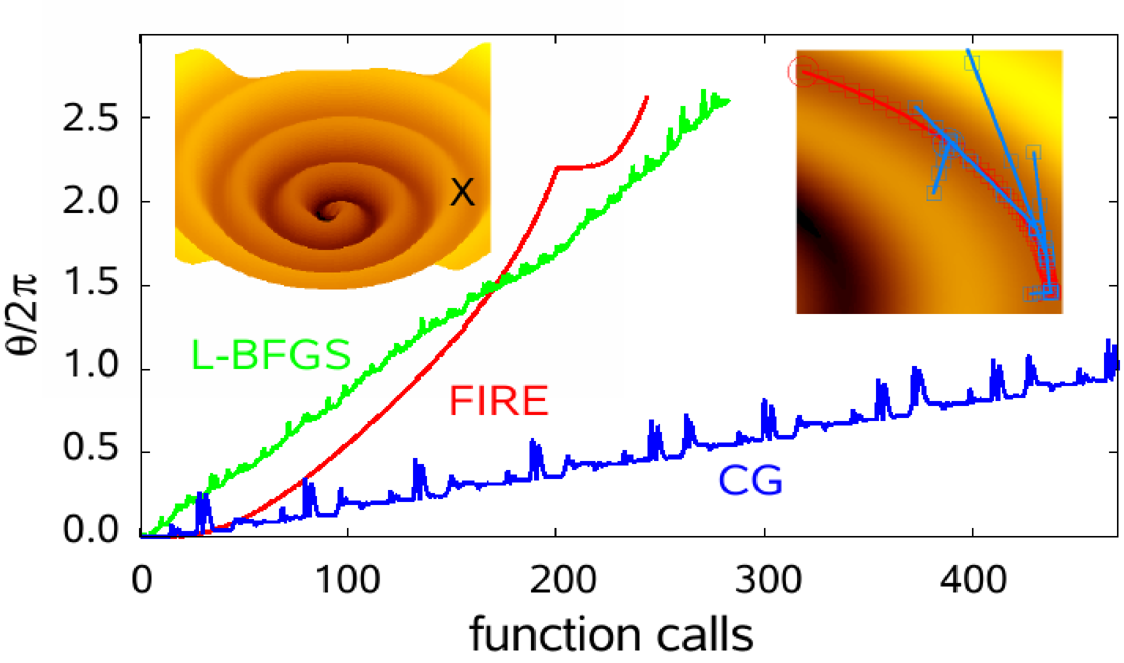
FIRE code

```
subroutine FIRE(it,dt,n,v[n],F[n],done)
  if( max(F) < crit ) done

  P = V*F
  V = (1-a)*V + a*F*norm(V)/norm(F)

  if( P<0 )
    V   = 0
    cut = it
    dt  = dt * f_dec
    a   = a_start
  else if( P>=0 and it-cut>N_min )
    dt  = min( dt*f_inc, dt_max )
    a   = a * f_dec
```

Have fun!

Do not click, clicking is not allowed.

Spiral



2.5
2.0
1.5
1.0
0.5
0.0
θ/2π
0
100
200
300
400
function calls
L-BFGS
FIRE
CG
X

FIRE, CG and L-BFGS