

Konstruksjon av høydimensjonale nevralt nettverk-potensialer for molekylærdynamikk

John-Anders Stende

Fysisk institutt
Universitetet i Oslo

Masterpresentasjon, oktober 2017

Table of Contents

Molekylærdynamikk

Nevrale nettverk

Nevralt nettverk-potensial

LAMMPS og TensorFlow

Konstruksjon av nevralt nettverk-potensial (NNP)

NNP for Si

Konklusjon og fremtidig arbeid

Hva er molekylærdynamikk?

- ▶ Numerisk metode for å simulere atomers og molekylers bevegelser i gasser, væsker og faste stoffer.
- ▶ Virtuelt eksperiment.

Dynamikk

- ▶ Partiklenes interaksjoner styrer dynamikken.
- ▶ Interaksjonene bestemmes av et kraftfelt \mathbf{F} :

$$\mathbf{F} = -\nabla V(\mathbf{r})$$

- ▶ Potensiell energiflate / potensial):

$$V(\mathbf{r}), \quad \mathbf{r} = (\mathbf{r}_1, \mathbf{r}_2, \dots, \mathbf{r}_N)$$

- ▶ $V(\mathbf{r})$ inneholder fysikken.

Ab initio molekylærdynamikk

Løse Schrödinger-likningen ved hvert tidssteg.

Klassisk molekylærdynamikk

Bruke en predefinert analytisk funksjon.

Klassisk potensial

$$V(\mathbf{r}) \approx \sum_i^N V_1(\mathbf{r}_i) + \sum_{i,j}^N V_2(\mathbf{r}_i, \mathbf{r}_j) + \sum_{i,j,k}^N V_3(\mathbf{r}_i, \mathbf{r}_j, \mathbf{r}_k) + \dots$$

1. Hvor mange ledd bør tas med?
2. Hvordan bør leddene se ut?

Eksperiementer / kvantemekanikk

Empirisk strategi:

1. Starte med en funksjonsform med noen parametre.
2. Bestemme parametre fra eksperimentelle data.

Kvantemekanisk strategi:

1. Produsere kvantemekaniske data.
2. Tilpasse en generell funksjonsform til datasettet.
 - ▶ Fordeler: Nøyaktig, ingen bias, overførbart.
 - ▶ Ulemper: Rent matematisk uttrykk, all relevant data må inkluderes.

Interpolere datasett

- ▶ Spliner
- ▶ Minste kvadraters metode
- ▶ Maskinlæring

Table of Contents

Molekylærdynamikk

Nevrale nettverk

Nevralt nettverk-potensial

LAMMPS og TensorFlow

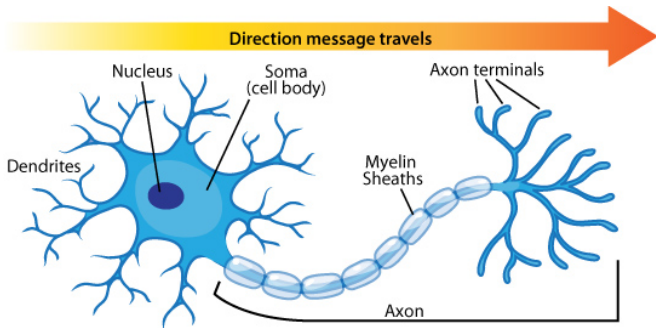
Konstruksjon av nevralt nettverk-potensial (NNP)

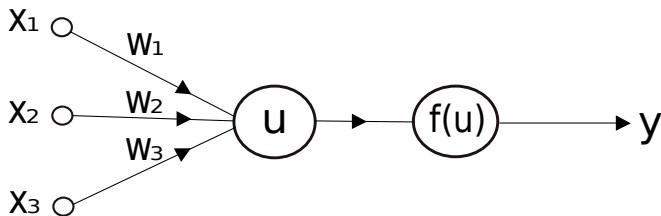
NNP for Si

Konklusjon og fremtidig arbeid

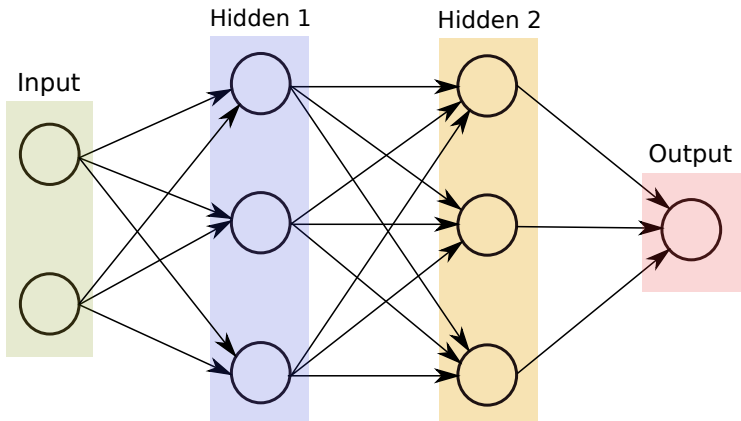
Kunstige nevrale nettverk

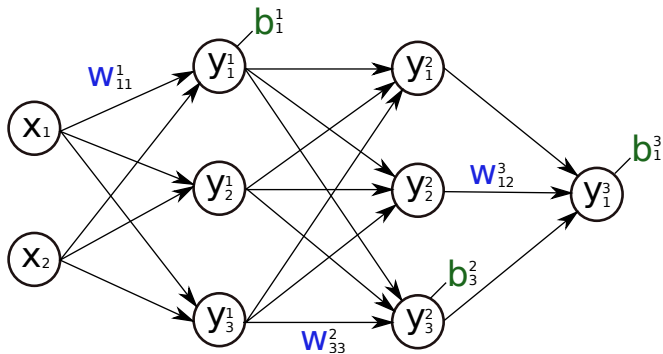
- ▶ Maskinlæringsalgoritme.
- ▶ Etterlikne en biologisk hjerne.
- ▶ Nettverk av matematiske nevroner.





$$y = f \left(\sum_{i=1}^n w_i x_i + b_i \right) = f(u)$$





$$y_i^1 = f_1(u_i^1) = f_1 \left(\sum_{j=1}^2 w_{ij}^1 x_j + b_i^1 \right)$$

$$y_1^3 = f_3 \left[\sum_{j=1}^3 w_{1j}^3 f_2 \left(\sum_{k=1}^3 w_{jk}^2 f_1 \left(\sum_{m=1}^2 w_{km}^1 x_m + b_k^1 \right) + b_j^2 \right) + b_1^3 \right]$$

$$= f_3(x_1, x_2)$$

- ▶ Dimensjonene til nettet må stemme overens med funksjonen som skal tilpasses.
- ▶ Det universelle approksimasjonsteoremet.

Aktiveringsfunksjoner

Restriksjoner

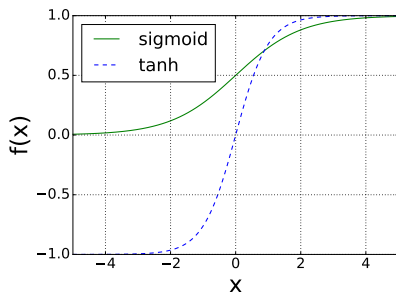
- ▶ Ikke-konstant
- ▶ Begrenset
- ▶ Monotont økende
- ▶ Kontinuerlig

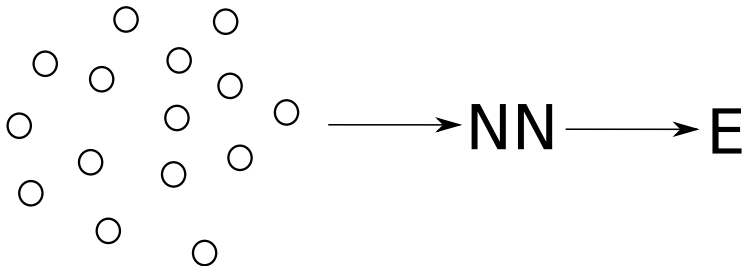
The sigmoid

$$f(x) = \frac{1}{1 + e^{-x}}$$

og hyperbolsk tangens

$$f(x) = \tanh(x)$$





Regresjon med nevralt nettverk

- ▶ Regresjon: Interpolere datasett $X \rightarrow Y$
- ▶ X : inputdata, Y : outputdata/målverdier.
- ▶ Trening: Iterativt minimere feilen til et sett med kjente energier.

Feilen defineres ved en cost-funksjon:

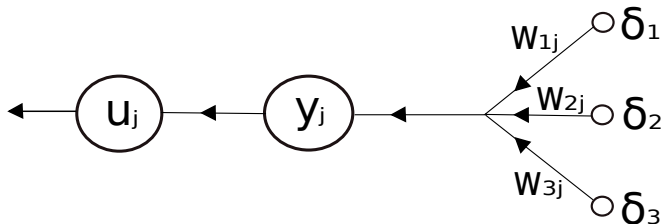
$$\Gamma = \frac{1}{2N} \sum_{i=1}^N (Y_i - y_i)^2$$

Optimering med gradient descent:

$$\theta_{k+1} = \theta_k - \gamma \nabla_{\theta_k} \Gamma(\theta)$$

Mange forskjellige måter å justere γ på.

Hvordan finne gradienten av nettet?



Backpropagation

1. Sender hvert treningsseksempel gjennom nettet.
2. Output (energi) sammenliknes med kjent energi.
3. Feilen propageres *bakover* ved kjerneregelen og vektene justeres.

Table of Contents

Molekylærdynamikk

Nevrale nettverk

Nevralt nettverk-potensial

LAMMPS og TensorFlow

Konstruksjon av nevralt nettverk-potensial (NNP)

NNP for Si

Konklusjon og fremtidig arbeid

Hvor mange nettverk?

- ▶ Ett nettverk: Uhåndterlig størrelse.
- ▶ Atomære nettverk,

$$E = \sum_{i=1}^N E_i$$

- ▶ Behler-Parrinello: Hver atomenergi E_i avhenger kun av naboatomer innenfor en cutoff-radius r_c .

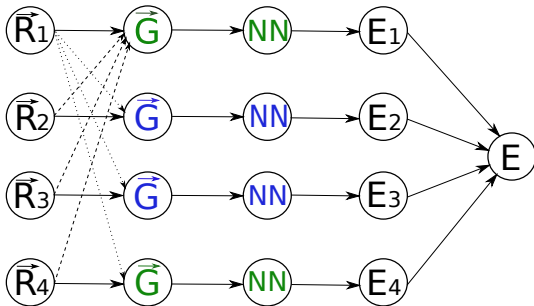
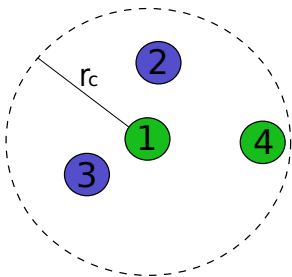
Høydimensjonalt potensial - utfordringer

1. Varierende antall naboer.
2. Translasjonell og rotasjonell invarians.
3. Rekkefølgen på naboer.

Behler-Parrinello-metoden: Symmetrifunksjoner

- ▶ Et sett av såkalte symmetrifunksjoner transformerer de kartesiske koordinatene til alle naboer til et gitt antall symmetriverdier.
- ▶ Funksjonene er definert som summer over naboer.

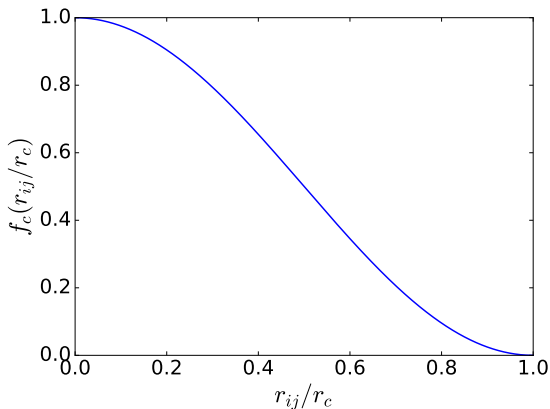
- ▶ Hvert atom har et eget nettverk og et sett av symmetrifunksjoner som beskriver cutoff-kulen.
- ▶ Hver atomtype har identiske nettverk og symmetrifunksjonssett.



Cutoff-funksjon

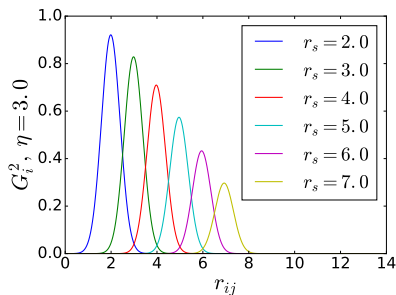
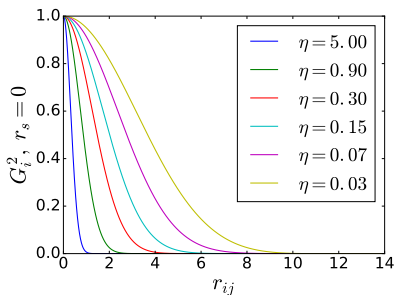
Monotonisk minkende del av en cosinus-funksjon.

$$f_c(r_{ij}) = \begin{cases} 0.5 \left[\cos \left(\frac{\pi r_{ij}}{r_c} \right) + 1 \right], & r_{ij} \leq r_c \\ 0, & r_{ij} > r_c \end{cases}$$



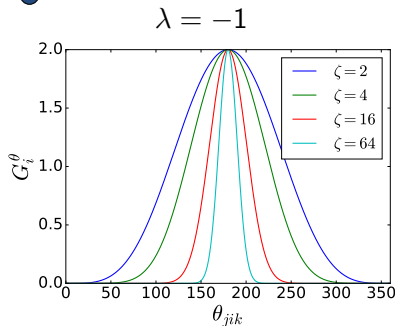
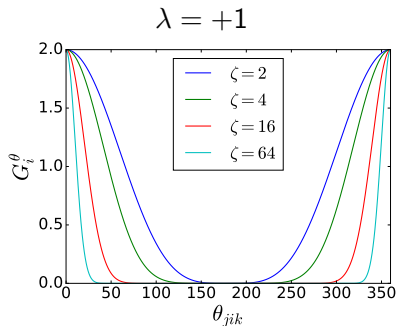
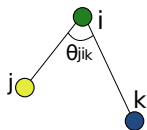
Radiell symmetrifunksjon

$$G_i^2 = \sum_{j=1}^N \exp[-\eta(r_{ij} - r_s)^2] f_c(r_{ij})$$



Angulær symmetrifunksjon

$$G_i^5 = 2^{1-\zeta} \sum_{j \neq i} \sum_{k > j} \left[(1 + \lambda \cos \theta_{jik})^\zeta \exp(-\eta(r_{ij}^2 + r_{ik}^2)) f_c(r_{ij}) f_c(r_{ik}) \right]$$



Krefter for et system av én atomtype

$$\mathbf{F}_i = -\nabla_i E$$

$$F_{i,x} = -\frac{\partial E}{\partial x_i} = -\sum_{j=1}^{N_i+1} \frac{\partial E_j}{\partial x_i} = -\sum_{j=1}^{N_i+1} \sum_{s=1}^M \frac{\partial E_j}{\partial G_s} \frac{\partial G_s}{\partial x_i}$$

$$E_j = \text{NN}[\mathbf{G}(\mathbf{r}_{jk})]$$

Table of Contents

Molekylærdynamikk

Nevrale nettverk

Nevralt nettverk-potensial

LAMMPS og TensorFlow

Konstruksjon av nevralt nettverk-potensial (NNP)

NNP for Si

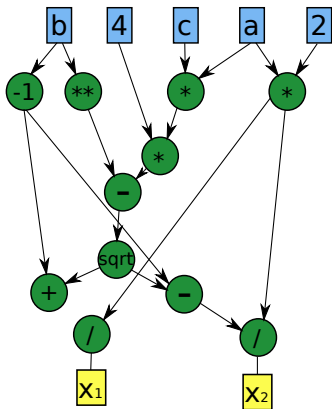
Konklusjon og fremtidig arbeid

LAMMPS

- ▶ Simuleringspakke for klassisk molekylærdynamikk utviklet ved Sandia National Laboratories.
- ▶ Kjøres ved inputscripts med egen syntaks.
- ▶ Vi har utvidet med samplingsalgoritme og nevral nettverk-potensial.

TensorFlow

- ▶ Maskinlæringspakke utviklet av Google.
- ▶ Setter opp nevrale nettverk som data flow-grafer (DFG).
- ▶ DFG: En graf bestående av *noder* som er forbundet av *kanter* med retning.
- ▶ Noder er matematiske operasjoner, kanter er tensorer.



$$ax^2 + bx + c = 0$$

$$x_1 = \frac{-b + \sqrt{b^2 - 4ac}}{2a}$$

$$x_2 = \frac{-b - \sqrt{b^2 - 4ac}}{2a}$$

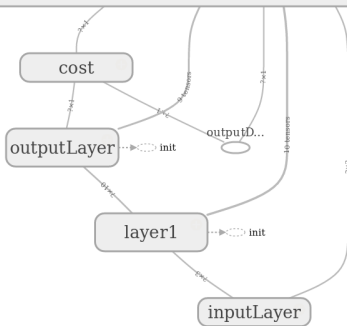
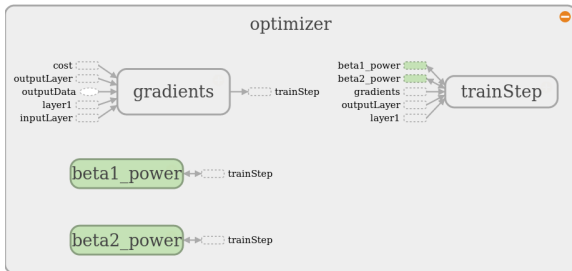


Table of Contents

Molekylærdynamikk

Nevrale nettverk

Nevralt nettverk-potensial

LAMMPS og TensorFlow

Konstruksjon av nevralt nettverk-potensial (NNP)

NNP for Si

Konklusjon og fremtidig arbeid

1. Generere treningsdata som er relevant for applikasjonen av NNP.
2. Trene et nevralt nettverk for å tilpasse en funksjon til dataene.
3. Bruke det trente nettverket som et analytisk potensial i molekylærdynamikksimuleringer.

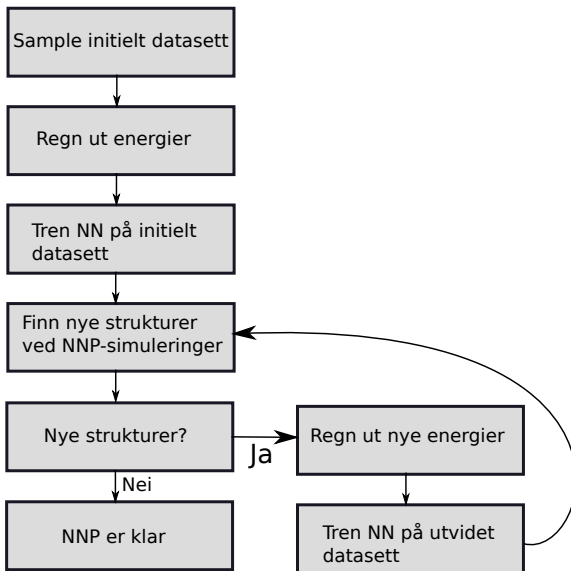


Table of Contents

Molekylærdynamikk

Nevrale nettverk

Nevralt nettverk-potensial

LAMMPS og TensorFlow

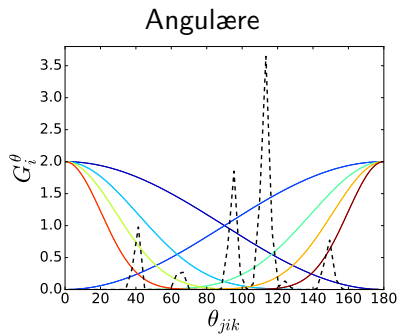
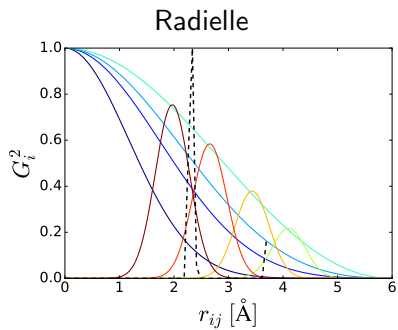
Konstruksjon av nevralt nettverk-potensial (NNP)

NNP for Si

Konklusjon og fremtidig arbeid

Initiell sampling

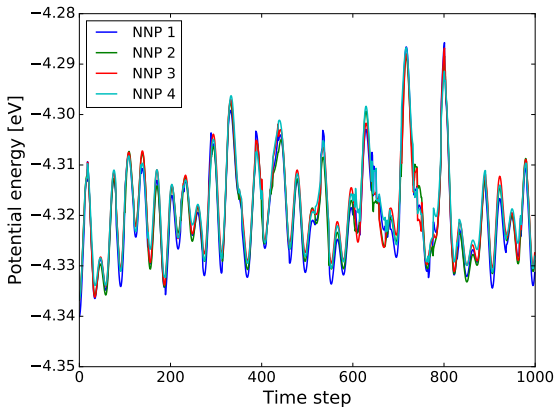
- ▶ Stillinger-Weber.
- ▶ $T \in [0, 500]$ K.
- ▶ Konfigurasjoner og energies samples med samplingsalgoritme.



Ekstrapolasjon

Lagre max og min for hver symmetryfunksjon.

Interpolasjon - multiple-NN-metoden



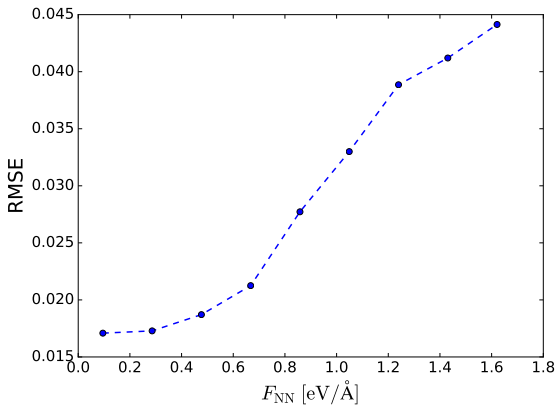
Gridsøk

Layers	Nodes	RMSE	Epoch	Time
$L = 1$	4	4.445	37035	575
	8	2.250	37305	570
	12	2.303	37980	622
	16	2.201	39780	630
	20	1.860	36180	617
	24	1.928	37305	621
	28	2.407	39375	697
	32	2.214	38700	672
$L = 2$	4	2.947	39960	750
	8	1.933	36180	671
	12	1.450	37350	766
	16	1.791	32265	633
	20	1.492	24840	546
	24	2.118	37620	819
	28	1.455	37350	895
	32	2.008	14895	344

Tilpasse endelig treningssett

Velger nettet med lavest RMSE etter 40000 epoker. RMSE: 0.864 meV.

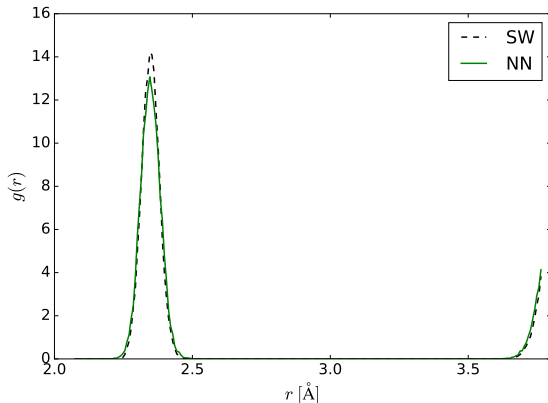
RMSE krefter



RMSE: 41.2 meV

Radiell distribusjonsfunksjon $g(r)$

Sammenlikner tidsmidlet SW og NN.



Mekaniske egenskaper

	NNP	Analytic SW	Relative error
Bulk modulus	103.0	101.4	1.58 %
Shear modulus	53.6	56.4	5.22 %
Poisson ratio	0.348	0.335	3.88 %

Table of Contents

Molekylærdynamikk

Nevrale nettverk

Nevralt nettverk-potensial

LAMMPS og TensorFlow

Konstruksjon av nevralt nettverk-potensial (NNP)

NNP for Si

Konklusjon og fremtidig arbeid

Det ideelle potensial

- ▶ Potensialet bør være nøyaktig.
- ▶ Burde finnes måter å systematisk forbedre potencialet.
- ▶ Potensialet bør være generelt og anvendbart på alle typer systemer.
- ▶ Potensialet bør kunne modellere faseoverganger.
- ▶ Potensialet bør være høydimensjonalt, dvs. avhenge av alle frihetsgrader.

Det ideelle potensial fortsetter

- ▶ Konstruksjonen av potensialet bør være så automatisert som mulig.
- ▶ Potensialet bør være prediktivt.
- ▶ Potensialet bør være raskt å evaluere.
- ▶ Konstruksjonen bør ikke ta for mye tid.
- ▶ Analytisk derivert bør være tilgjengelig.

Fremtidig arbeid

- ▶ *Ab initio* data.
- ▶ Mer nøyaktige krefter.
- ▶ Andre systemer.
- ▶ Optimering.