Lessons Learned 1, 1&2 (2021)
Submitted: 28.05.2021
Accepted: 09.06.2021
DOI: https://doi.org/10.25369/ll.v1i1/2.17
ISSN: 2749-1293 (Print); 2749-1307 (Online)

Virtuelle PC Pools für Computerpraktika am Beispiel der Materialwissenschaften

S. Kampmann[1], D. Bodesheim[1], A. Croy[1], T. Schied[1], R. Gutierrez[1], A. Dianat[1], G. Cuniberti[1,2,3,*]

[1] *Lehrstuhl für Materialwissenschaft und Nanotechnik, Institut für Werkstoffwissenschaft, Fakultät Maschinenwesen, Technische Universität Dresden*
[2] *Dresden Center for Computational Materials Science (DCMS), Technische Universität Dresden*
[3] *Dresden Center for Intelligent Materials (DCIM), TU Dresden*

Abstract

Computerpraktika stellen einen wichtigen Bestandteil vieler Lehrveranstaltungen dar, welche die Grundlagen und Details von computergestützten Methoden vermitteln sollen. In den Materialwissenschaften spielen solche Methoden eine zunehmend wichtige Rolle. Typischerweise setzen die Praktika eine physische Präsenz in den PC Pools voraus, u.a. da eine Vielzahl von verschiedenen Programmen lokal installiert und bereitgestellt werden muss. Um Computerpraktika auch in der Online-Lehre vollumfänglich und weitestgehend unabhängig von den Gegebenheiten der Studierenden einsetzen zu können, wurde im Wintersemester 2020/21 ein virtueller PC Pool auf Basis von virtuellen Maschinen mit Web-basiertem Zugang eingerichtet. Dieser virtuelle PC Pool wurde in verschiedenen Lehrveranstaltungen erfolgreich eingesetzt und kann auch bei hybriden Lehrformaten in verschiedenen Disziplinen verwendet werden.

Computer practicals are an important part of many courses that are designed to teach the basics and details of computer-based methods. In the materials sciences, such methods play an increasingly important role. Typically, the practical courses require a physical presence in the PC pools, among other things because a large number of different programmes have to be installed and made available locally. In order to be able to use computer practicals fully and as far as possible independently of the students' circumstances in online teaching, a virtual PC pool based on virtual machines with web-based access was set up in the winter semester 2020/21. This virtual PC pool has been successfully used in various courses and can also be used in hybrid teaching formats in various disciplines.

*Corresponding author: gianaurelio.cuniberti@tu-dresden.de

1. Einleitung

In den meisten Lehrveranstaltungen mit Bezug zu Computersimulationen sind Praktika vorgesehen, durch die die Studierenden die Simulationsmethoden ausprobieren und anwenden sollen. Die Computerpraktika finden typischerweise in PC Pools statt, wo die notwendige Software zur Verfügung steht und Fragen durch den Praktikumsleiter direkt vor Ort beantwortet werden können. Außerhalb der Pools können die Studierenden die entsprechende Software zwar prinzipiell auf dem eigenen PC installieren, was sich aber durch die Heterogenität der verschiedenen Hardware und Betriebssysteme teilweise sehr schwierig gestaltet. Ein virtueller PC Pool, bei dem die benötigte Umgebung serverseitig bereitgestellt wird, bietet eine praktikable Lösung für die digitale Lehre. Aber auch hier sind verschiedene Aspekte, insbesondere bezüglich eines sicheren und niedrigschwelligen Zugangs, zu beachten [1-3].

Im vorliegenden Artikel soll am Beispiel der Lehrveranstaltungen *Computersimulation in der Materialwissenschaft*, *Computational Methods II*, *Concepts of Molecular Modeling, Computational Materials Science*: *Molekulardynamik* und *Kontinuumsmethoden*, welche für Studierende der *Werkstoffwissenschaft* im 8. und 9. Semester, sowie für Studierende verschiedener Masterstudiengänge (*Computational Modelling and Simulation*, *Nanobiophysics*, *Organic and Molecular Electronics*, sowie *Physik*) an der TU Dresden angeboten werden, die Einrichtung und Nutzung eines virtuellen PC Pools für die Materialwissenschaften beschrieben werden. Die hier vorzustellende Lösung wurde mit Hilfe des Zentrums für Informationsdienste und Hochleistungsrechnen (ZIH) der TU Dresden erarbeitet und umgesetzt.

2. Anforderungen

Wie eingangs erwähnt, sollen die Praktika den Vorlesungsstoff durch eigene Versuche ergänzen und mit Hilfe von Projekten vertiefen. Hierfür werden den Studierenden Aufgaben gestellt, welche mit bereitgestellter Software bearbeitet werden sollen. Die Simulationsergebnisse müssen anschließend ausgewertet und interpretiert werden. Im Rahmen der Praktika mit Bezug zur Materialwissenschaft wird verschiedene (freie) Software zur Materialsimulation unter Linux eingesetzt. Hierbei müssen verschiedene Größenskalen der zu untersuchenden Materialien abgedeckt und mit der eingesetzten Software behandelt werden können (siehe Abb. 1 und Tab. 1).

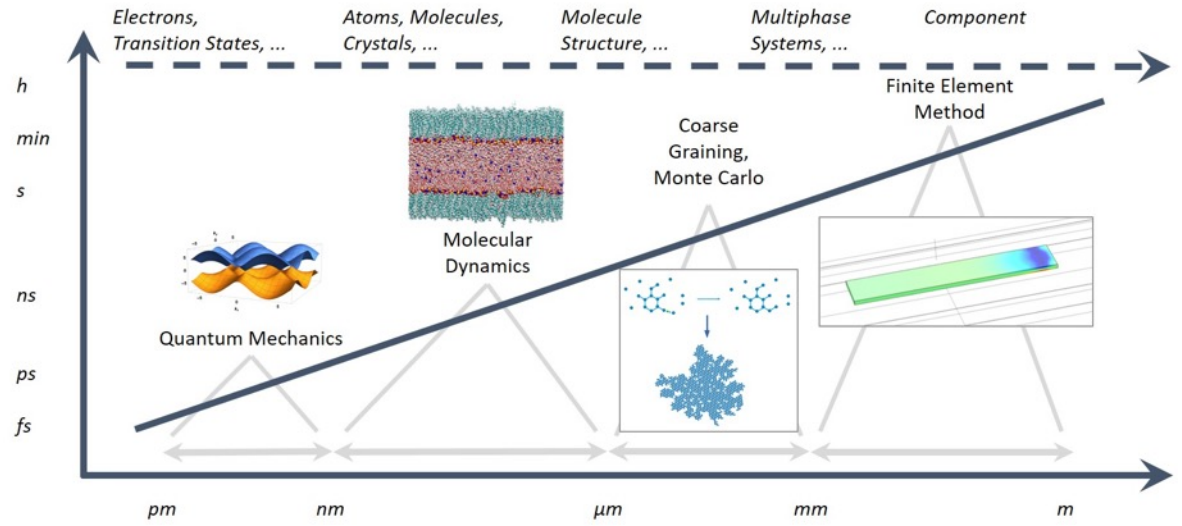


Abb. 1: Übersicht der Methoden zur Beschreibung verschiedener Zeit- und Längenskalen.

Tab. 1: In den Computerpraktika verwendete Softwareprogramme und deren Anwendungsbereich (Skala)

Software	Beschreibung	**Atomistisch**	**Mikro**	**Meso**	**Makro**
Jupyter-Notebook [15]	Auswertung und Visualisierung	☑	☑	☑	☑
OVITO [12]	3D Visualisierung und Analyse	☑	☑	☒	☒
VMD [16]	3D Visualisierung und Analyse	☑	☑	☒	☒
LAMMPS [11]	Molekulardynamik Simulationen	☑	☑	☒	☒
DFTB+ [13]	Elektronenstrukturberechnungen	☑	☒	☒	☒
Avogadro 2 [17]	Visualisierung von Molekülen und deren Manipulation	☑	☒	☒	☒
COMSOL [14]	Finite Elemente Software	☒	☒	☑	☑

Diese Programme sollen auch im virtuellen PC Pool allen Studierenden zur Verfügung stehen. Zusätzlicher Administrationsaufwand durch Installationssupport auf verschiedenen Hardwareplattformen und Betriebssystemen sollte weitestgehend vermieden werden. Idealerweise sollen die Studierenden auch außerhalb der Praktikumszeiten vollen Zugriff auf Programme, Daten und Rechnerkapazitäten haben. Zusätzlich sollte darauf geachtet werden, dass die Studierenden identische virtuelle Desktopumgebungen bereitgestellt bekommen. Zudem ist insgesamt der Datenschutz zu gewährleisten.

3. Umsetzung

Für die Umsetzung des virtuellen PC Pools, der eine Kapazität von 80 virtuellen Maschinen umfasste, wurde eine Lösung erarbeitet, die auf den folgenden Komponenten beruht:

- Die virtuellen Maschinen wurden in einer Cloud-Umgebung des ZIH betrieben. Die Konfiguration sowie das Monitoring der virtuellen Maschinen erfolgt über ein Web-Interface, das ebenfalls vom ZIH bereitgestellt wird.
- Jede virtuelle Maschine basierte auf einer vordefinierten Installation, welche sämtliche Software umfasste. Als Betriebssystem wurde *Ubuntu 20.04* [4] verwendet.
- Als Virtual Network Computing (VNC) Server kam *Turbo-VNC* zum Einsatz [5]. Der VNC Server ermöglicht den Fernzugriff auf die virtuelle Maschine.
- Der Zugang wurde über ein Web-Interface mit Hilfe der Software *noVNC* realisiert [6] und durch Transport Layer Security (TLS) abgesichert. Als ressourcenschonende Desktopoberfläche wurde *Xfce* [7] gewählt (siehe Abb. 2).
- Den virtuellen Maschinen wurde jeweils eine feste Domain zugewiesen.

Aus Sicht der Studierenden ergeben sich mit dieser Lösung eine Reihe von Vorteilen. Jeder Nutzer hat eine eigene, individuelle virtuelle Umgebung. Der Zugriff erfolgt über den Browser und ist daher weitestgehend hardware- und betriebssystemunabhängig. Auch bisher IT-Unerfahrene können unproblematisch Zugang auf einer Vielzahl von Endgeräten erhalten. Die Übertragung ist verschlüsselt und der Zugang ist passwortgeschützt. Sämtliche Software ist vorinstalliert und getestet, wodurch allen Teilnehmenden identische Software zur Verfügung steht. Es können auch lizenzpflichtige Programme ohne eigene lokale Installa-

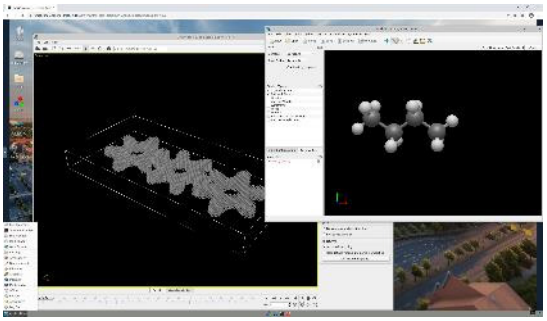


Abb. 2: Screenshot des virtuellen Desktops. Im Vordergrund sind Visualisierungen mit OVITO und Avogadro 2 zu sehen.

tion genutzt werden, da der Bereich der IP Adressen eingeschränkt ist. Daten können über eine Cloudlösung oder durch Netzlaufwerke bereitgestellt und abgerufen werden.
Aus Administratorensicht bietet die gefundene Lösung ebenfalls mehrere Vorteile. Die im Praktikum verwendete Software kann weitestgehend zentralisiert verwaltet und getestet werden. Die Ressourcen können dynamisch an den Bedarf angepasst werden und sind prinzipiell immer verfügbar. Der Zugang und die Verwaltung können dabei graphisch oder per Terminal erfolgen. Für eine zentrale Administration kann die Software *Cluster-SSH* [8] verwendet werden, mit der man gleichzeitig auf alle virtuellen Maschinen zugreifen und Befehle ausführen kann. Auf diese Art und Weise wurden auch randomisierte Passwörter, die für den Webzugriff erstellt wurden, den einzelnen Maschinen zugewiesen. Auch ist somit die Nachinstallation von Software oder das Ablegen von Dateien komfortabel möglich. Zudem ist auf diesem Wege die Rückführung der virtuellen Maschinen in den Ausgangszustand nach Semesterende problemlos möglich.

4. Durchführung

An den Lehrveranstaltungen *Computersimulation in der Materialwissenschaft*, *Computational Methods* und *Concepts of Molecular Modeling* nahmen im Wintersemester 2020/21 insgesamt 80 Studierende teil. Jeder Teilnehmende bekam seine eigene virtuelle Maschine zugewiesen. Zusätzlich wurde den Mitarbeitern, die die Praktika betreuen, jeweils eine virtuelle Maschine, die identisch mit den anderen virtuellen Maschinen war, zur Verfügung gestellt. Während der online-Praktikumsveranstaltungen konnten die Studierenden mit Hilfe der Bildschirmübertragungsfunktion gängiger Videokonferenzsysteme, wie z.B. Big Blue Button oder Zoom, die Arbeitsschritte auf der grafischen Oberfläche der virtuellen Maschine nachvollziehen. Durch die einheitliche Installation konnten auftretende Fehler schnell eingegrenzt werden.
Zur Bereitstellung praktikumsrelevanter Dateien wurde die Lernplattform OPAL [9] und der von der TU Dresden bereitgestellte Cloudservice [10] verwendet. Der Zugang zu diesen Ressourcen auf der virtuellen Maschine ist wie gewohnt über den Browser möglich.
Insbesondere für die Vorlesung *Concepts of Molecular Modelling* war der einfache und unbeschränkte Zugriff sehr wertvoll, da es Studierende aus den internationalen Masterstudiengängen gab, die sich nicht in Deutschland aufhielten und damit in vielen Fällen nur asynchron Zugang zu Programmen und Daten hatten.

In den Praktika dieser Lehrveranstaltung sollen die Studierenden den Umgang mit Molekulardynamikprogrammen wie *LAMMPS* [11], mit Visualisierungssoftware wie *OVITO* [12] und mit der Programmiersprache Python zur Datenauswertung und zur Durchführung von Monte Carlo Simulationen lernen (siehe Abb. 3). Die gelernten Fähigkeiten wurden am Ende des Praktikums in einem kleinen von den Studierenden durchgeführten Projekt über Molekulardynamik oder Monte Carlo angewandt. Schwerpunkte der Lehrveranstaltung *Computational Methods* sind Methoden zur Berechnung der elektronischen Struktur, welche am

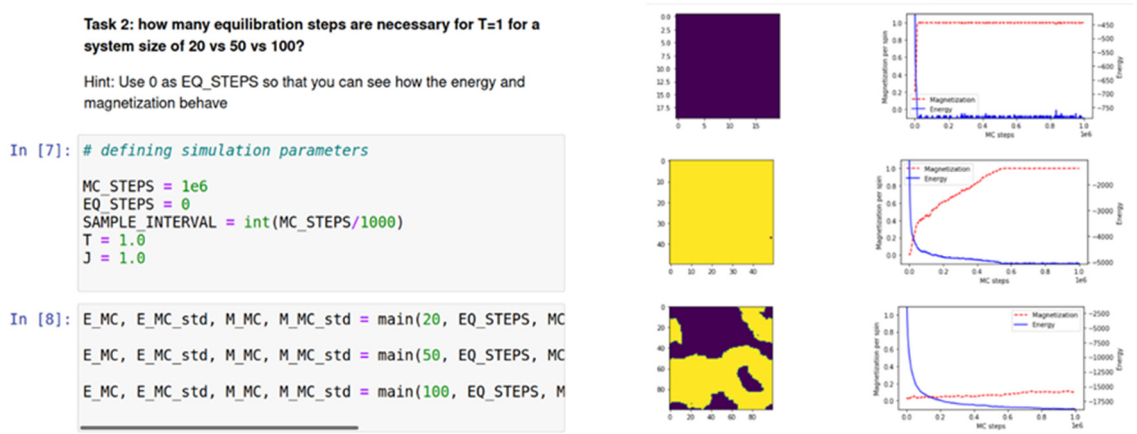


Abb. 3: Auszug aus einem Jupyter-Notebook über Monte Carlo Simulationen.

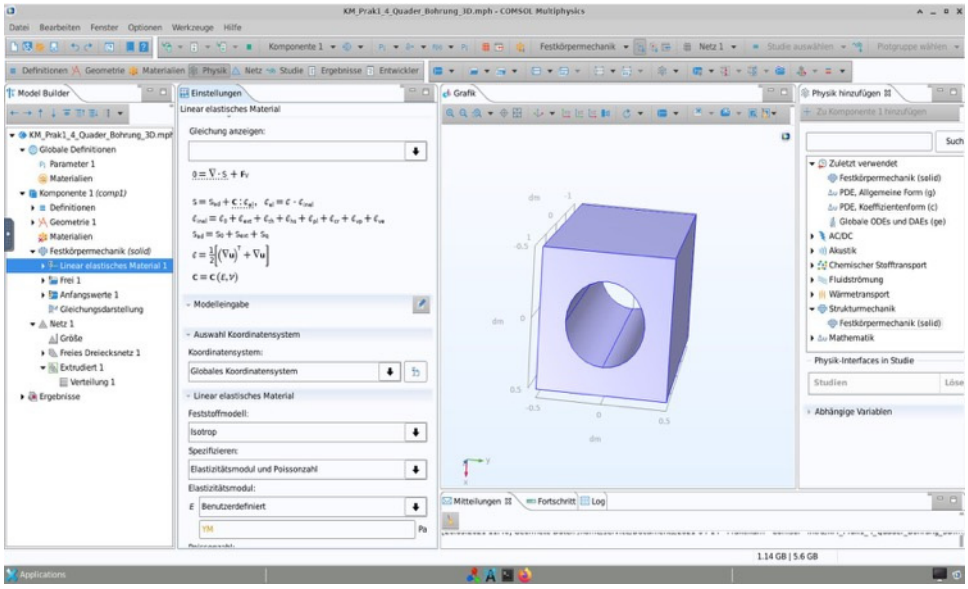


Abb. 4: Screenshot eines Berechnungsbeispiels in der Finite Elemente Software COMSOL [14] aus dem Praktikum der Lehrveranstaltung Kontinuumsmethoden.

Beispiel von *DFTB+* [13] praktisch erprobt werden. Bei der *Density Functional Based Tight Binding* (DFTB) Methode handelt es sich um eine Näherung der Dichtefunktionaltheorie, die insbesondere eine Betrachtung von sehr großen Systemen erlaubt. Im Praktikum der Lehrveranstaltung *Kontinuumsmethoden* werden grundlegende Stofftransportphänomene, sowie thermische bzw. mechanische Problemstellungen behandelt. Gleichzeitig wird der praktische Umgang mit der weit verbreiteten (lizenzpflichtigen) Finite Elemente Software *COMSOL* [14] anhand von Berechnungen vereinfachter Beispiele solcher Problemstellungen vermittelt (siehe Abb. 4). Gerade bei der Verwendung lizenzpflichtiger Software bietet der virtuelle PC Pool eine einfache Lösung, allen Studierenden einen Zugang zu ermöglichen. Das Verständnis der Herangehensweise an materialwissenschaftliche Fragestellungen und deren Bearbeitung mit Hilfe von Kontinuumsmethoden befähigt die Teilnehmenden schließlich, sich Lösungen anders gearteter Probleme, die nicht Teil des Lehrumfangs sind, selbstständig zu erschließen.

In den folgenden zwei Unterkapiteln werden der Aufbau und Ablauf jeweils einer Praktikumseinheit exemplarisch dargestellt. Dabei wird zunächst auf die Verwendung von Python eingegangen und danach eine Fragestellung mit Bezug auf die Molekulardynamik beschrieben.

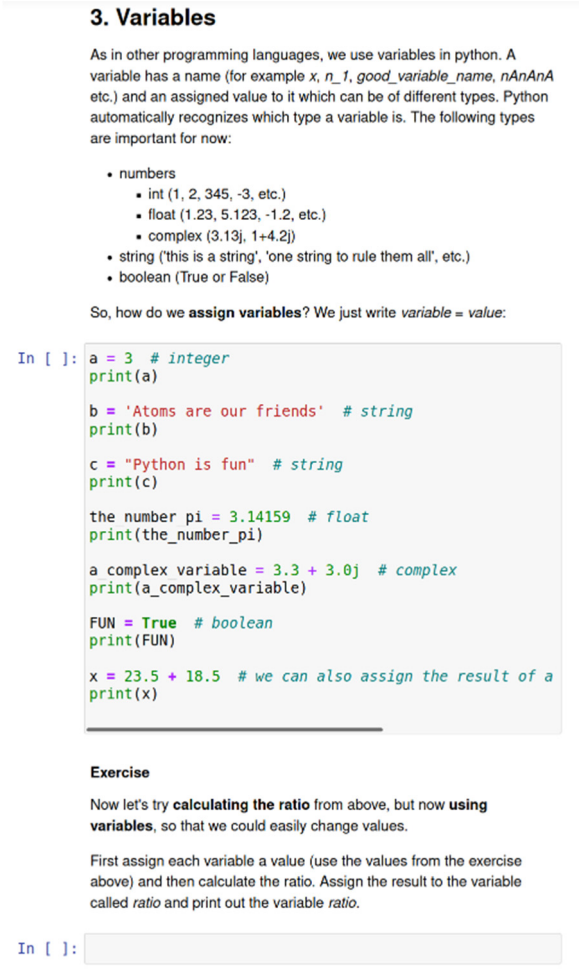
4.1 Praktikum 1: Python

Am Anfang des Semesters erfolgte eine Einführung in Python, um auch Studierenden ohne jegliche Vorkenntnisse in Programmierung einen leichten Einstieg zu ermöglichen. Ziel war, dass die Studierenden einfachen Python-Code verstehen und ebenfalls selbstständig Skripte, zum Beispiel zur Datenauswertung, erstellen können. Daher wurde bei den Kursen auf möglichst große Beteiligung und Eigenarbeit der Studierenden gesetzt, um einen möglichst großen Lernerfolg zu garantieren.

Die Tutorien wurden online über die Plattform Zoom abgehalten. Die Studierenden mussten eigenständig Programmieraufgaben während des Tutoriums lösen, welche dann besprochen wurden. Es wurde regelmäßig über Umfragen der aktuelle Fortschritt der Studierenden bei der Aufgabenbearbeitung geprüft, um zu ermitteln, wann welche Aufgaben besprochen werden sollten. Des Weiteren wurden die Tutorien aufgezeichnet, so dass asynchrones Lernen problemlos möglich war, falls zum Beispiel Studierende in einer anderen Zeitzone waren oder sonstige andere Verpflichtungen hatten.

Die Programmieraufgaben wurden in sogenannten *Jupyter-Notebooks* [15] erstellt. Dies ist eine Umgebung, in welcher Abschnitt für Abschnitt Code-Teile ausgeführt werden können und ebenfalls Textbausteine eingefügt werden können (siehe Abb. 5). Diese Jupyter-Note-

books wurden so gestaltet, dass genaue Anweisungen und Erklärungen bereits enthalten waren und kleine Programmieraufgaben zu lösen waren. Die Studierenden sollten dann während des Kurses die Aufgaben in den Notebooks eigenständig lösen, während Tutoren in der Zoom-Konferenz für Fragen bereitstanden. Die Notebooks konnten in den virtuellen Maschinen über OPAL heruntergeladen und bearbeitet werden.



3. Variables

As in other programming languages, we use variables in python. A variable has a name (for example *x*, *n_1*, *good_variable_name*, *nAnAnA* etc.) and an assigned value to it which can be of different types. Python automatically recognizes which type a variable is. The following types are important for now:

- numbers
  - int (1, 2, 345, -3, etc.)
  - float (1.23, 5.123, -1.2, etc.)
  - complex (3.13j, 1+4.2j)
- string ('this is a string', 'one string to rule them all', etc.)
- boolean (True or False)

So, how do we **assign variables**? We just write *variable = value*:

In [ ]:
```
a = 3  # integer
print(a)

b = 'Atoms are our friends'  # string
print(b)

c = "Python is fun"  # string
print(c)

the number pi = 3.14159  # float
print(the_number_pi)

a complex variable = 3.3 + 3.0j  # complex
print(a_complex_variable)

FUN = True  # boolean
print(FUN)

x = 23.5 + 18.5  # we can also assign the result of a
print(x)
```

Exercise

Now let's try **calculating the ratio** from above, but now **using variables**, so that we could easily change values.

First assign each variable a value (use the values from the exercise above) and then calculate the ratio. Assign the result to the variable called *ratio* and print out the variable *ratio*.

In [ ]:

Abb. 5: Auszug aus einem Jupyter-Notebook zur Einführung in die Programmierung mit Python.

Neben den Einführungskursen zu Python gab es außerdem ein Python-basiertes Tutorium über Monte-Carlo Simulationen (siehe Abb. 3). Dieses Tutorium wurde ebenfalls in Jupyter-Notebooks abgehalten, indem auch hier Erklärungen und Aufgaben enthalten waren, die von den Studierenden bearbeitet wurden.

Am Ende des Semesters bearbeiteten die Studierenden ein eigenes kleines Projekt, über welches Sie einen Bericht anfertigten. Da die Virtuellen Maschinen jederzeit über den Browser von zu Hause aus nutzbar waren, konnten alle Studierenden selbstständig von zu Hause dieses Projekt durchführen.

4.2 Praktikum 2: Molekulardynamik

Molekulardynamik-Simulationen sind eine wichtige Säule der computergestützten Materialwissenschaft. Im Rahmen der Praktika lösten die Studierenden vorgegebene Aufgaben anhand von kommentierten Eingabeskripten. Eine solche Aufgabenstellung lautet z.B.

1) Generieren Sie ein ideales Gold-Nanowire (kfz, L = 8 nm, R = 1,2 nm).
2) Berechnen Sie die Abhängigkeit des Druckes von der Dehnung in z-Richtung.
3) Berechnen Sie die Abhängigkeit der potentiellen Energie von der Dehnung während des Dehnungsprozesses und berechnen Sie die Bruchdehnung für das Nanowire.

Die Ausführung des entsprechenden Skripts dauert nur wenige Minuten. Anschließend kann das Verhalten des Nanowire mit Hilfe von *OVITO* visualisiert und so die kritische Dehnung gefunden werden (siehe Abb. 6). Die Auswertung der Druck-Dehnung- und Energie-Dehnung-Kurven kann dann beispielsweise in einem Jupyter-Notebook oder durch separate Software erfolgen. Insgesamt fanden vier Praktika statt, in denen jeweils eine Aufgabe, wie exemplarisch beschrieben, bearbeitet wurde. Thematisch wurden dabei unter anderem folgende Themengebiete abgearbeitet: Generierung und Deformation von Kohlenstoffstrukturen; Aufwärm- und Abkühlprozesse von Nanostrukturen.

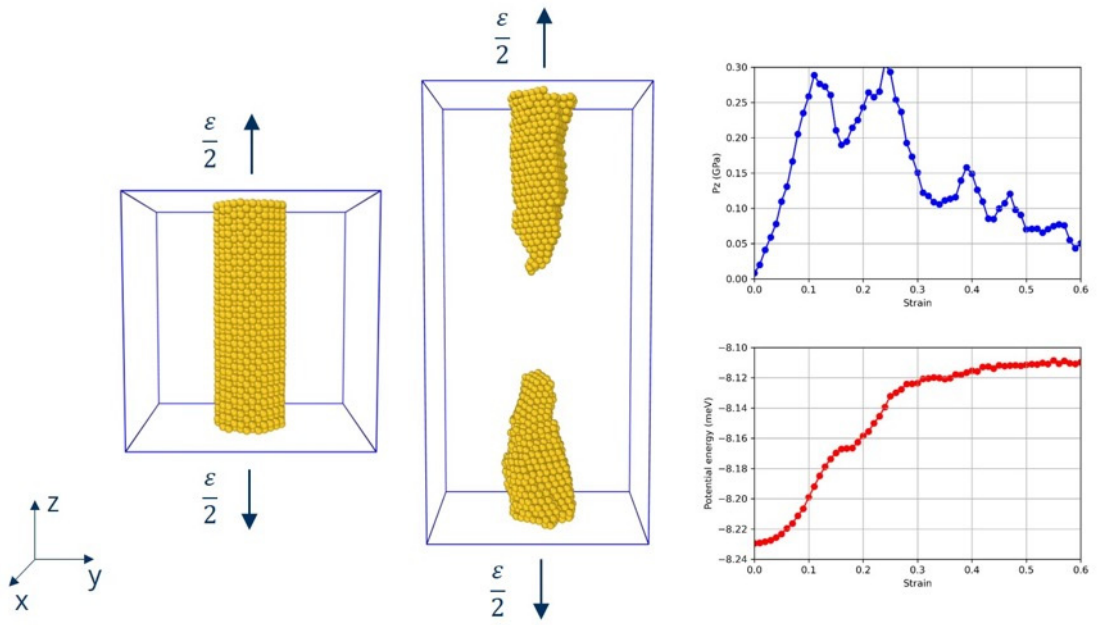


Abb. 6: Visualisierung eines Gold-Nanowire in OVITO a) ohne Dehnung, und b) mit 60 Prozent Dehnung. c) Mechanische Spannung (oben) und potentielle Energie (unten) in Abhängigkeit von der Dehnung

5. Fazit und Ausblick

Insgesamt stellt die hier vorgestellte Lösung eine komfortable und praktikable Möglichkeit für die Durchführung von Computerpraktika auch im Bereich der hybriden Lehre dar. Durch den Zugang über die virtuellen Maschinen konnten die Praktika ohne Einschränkungen durchgeführt werden. Dabei ist die standardisierte Bereitstellung von Software und der sichere, jederzeitige und unproblematische Zugang durch die Weboberfläche hervorzuheben. In Zukunft besteht daher die Möglichkeit die Aufgaben und Lösungsfortschritte der Studierenden parallel in den Computerpools und den virtuellen Maschinen zur Verfügung zu stellen. Somit kann der flexible Zugang für die Studierenden zu den Praktika-Ressourcen gewährleistet und gleichzeitig die individuelle Lehrbetreuung gestärkt werden.

Danksagungen

Wir möchten uns insbesondere bei Herrn Holger Mickler, Herrn Christopher Mosch und Herrn Matthias Jurenz vom ZIH der TU Dresden für Ihre sehr hilfreiche Unterstützung bei der Konzeptfindung, Installation und Fehleranalyse bedanken.

Literatur

[1] Luo, F., Gu, C. and Li, X. 2015. Constructing a virtual computer laboratory based on Open Stack. *10th International Conference on Computer Science & Education (ICCSE). doi: 10.1109/ICCSE.2015.7250353*

[2] Bastidas, C. E. C. 2011. Enabling remote access to computer networking laboratories for distance education. *2011 Frontiers in Education Conference (FIE). doi: 10.1109/FIE.2011.6142731*

[3] Hu, X., Le, H., Bourgeois, A. G. and Pan, Y. 2018. Collaborative Learning in Cloud-based Virtual Computer Labs. *2018 IEEE Frontiers in Education Conference (FIE). doi: 10.1109/FIE.2018.8659018*

[4] https://releases.ubuntu.com/20.04/

[5] https://www.turbovnc.org/

[6] https://novnc.com/

[7] https://www.xfce.org/

[8] https://github.com/duncs/clusterssh

[9] https://bildungsportal.sachsen.de/opal

[10] https://tu-dresden.de/zih/dienste/service-katalog/zusammenarbeiten-und-forschen/datenaustausch/cloudstore

[11] https://lammps.sandia.gov/

[12] https://www.ovito.org/

[13] https://dftbplus.org/

[14] https://www.comsol.de/

[15] https://jupyter.org/

[16] http://www.ks.uiuc.edu/Research/vmd/

[17] https://www.openchemistry.org/projects/avogadro2/