

Tutorial Prático de VASP

Cálculo de silício bulk no cluster Heisenberg

Material de aula — Ilum / CNPEM

Objetivos da aula

Ao fim deste tutorial, vocês saberão:

- Como montar os quatro arquivos de entrada do VASP (INCAR, POSCAR, KPOINTS, POTCAR) para um sistema bulk.
- Como construir o POTCAR concatenando os pseudopotenciais corretos.
- Como submeter um cálculo no cluster heisenberg.cnpem.br via SLURM.
- Como interpretar os principais arquivos de saída e verificar a convergência.
- Como aplicar isso à célula primitiva do silício e extrair a energia total e o parâmetro de rede otimizado.

1. O sistema: silício na estrutura diamante

O silício cristalino é o sistema didático por excelência da estrutura eletrônica de sólidos. Ele tem todas as propriedades que tornam um exemplo ideal: estrutura bem conhecida (cúbica diamante), poucos átomos na célula primitiva (apenas 2), é semicondutor (então não precisamos nos preocupar com smearings sofisticados para metais), e o parâmetro de rede experimental é precisamente conhecido (3,57 Å a 25 °C).

A estrutura diamante é uma rede FCC com dois átomos de silício por célula primitiva: um em (0, 0, 0) e outro em (1/4, 1/4, 1/4) em coordenadas diretas. A célula convencional cúbica tem oito átomos, mas é mais eficiente usar a célula primitiva — o cálculo fica oito vezes mais barato e dá a mesma física.

Dica: Quanto menor for a célula unitária, mais barato o cálculo. Sempre utilize a célula primitiva quando o que vocês querem é só a estrutura eletrônica. Utilize a célula convencional apenas se precisarem da forma cúbica explícita (por exemplo, ao criar supercélulas).

2. Preparação do ambiente no Heisenberg

Vocês já sabem se conectar ao heisenberg.cnpem.br via SSH e navegar pelo sistema de arquivos a partir das aulas de ORCA. Comecem criando um diretório dedicado para este cálculo:

```
ssh seu_usuario@heisenberg.cnpem.br
cd $SCRATCH      # ou onde vocês costumam trabalhar
mkdir -p vasp_tutorial/si_bulk
cd vasp_tutorial/si_bulk
```

É boa prática ter um **diretório separado para cada cálculo**. Como o VASP escreve muitos arquivos de saída com nomes fixos, misturar cálculos diferentes na mesma pasta é receita para confusão e perda de dados. Vejam que no ORCA vocês mudavam os nomes de arquivo e conseguiam rodar vários cálculos na mesma pasta. **Isso não funcionará com o VASP.**

3. O arquivo INCAR

O INCAR é o cérebro do cálculo. Cada linha tem o formato TAG = valor, com comentários após !. O VASP é tolerante com ordem e espaços, mas atenção ao tipo de cada valor: inteiros, reais com ponto decimal, e booleanos sempre como .TRUE. ou .FALSE. (com os pontos).

Para o nosso silício bulk, criem o arquivo INCAR com o seguinte conteúdo:

```
SYSTEM = Si bulk - celula primitiva diamante
```

```

! Inicializacao
ISTART = 0          ! comecar do zero
ICHARG = 2          ! densidade inicial = soma de densidades atomicas

! Base de ondas planas
ENCUT = 400         ! eV (convergir depois!)
PREC = Accurate     ! grade FFT precisa

! Convergencia SCF
EDIFF = 1E-06       ! eV
NELM = 60           ! maximo de iteracoes eletronicas
ALGO = Normal       ! algoritmo Davidson (robusto)

! Smearing - Si e semiconductor
ISMEAR = 0          ! Gaussiano
SIGMA = 0.05        ! eV (pequeno para semiconductor)

! Projecao PAW
LREAL = .FALSE.     ! espaco reciproco (celula pequena)

! Tipo de tarefa: apenas SCF, sem mover atomos
NSW = 0
IBRION = -1

! O que salvar
LWAVE = .FALSE.
LCHARG = .FALSE.

```

Cada uma dessas tags merece atenção. A tabela a seguir explica cada uma:

Tag	Valor no exemplo	Significado
SYSTEM	Si bulk	Apenas um rótulo descritivo, ignorado pelo VASP. Útil para identificar o cálculo nos logs.
ISTART	0	Como iniciar o cálculo. 0 = começar do zero (sem WAVECAR prévio). 1 = ler WAVECAR existente.
ICHARG	2	Densidade inicial. 2 = superposição de densidades atômicas (padrão para sistemas novos).
ENCUT	400	Energia de corte das ondas planas em eV. Controla a qualidade da base — deve ser convergido.
PREC	Accurate	Precisão geral (Normal, Accurate, High). 'Accurate' é o padrão recomendado para produção.
EDIFF	1E-06	Critério de convergência eletrônica (SCF) em eV. 10^{-6} é adequado para energias; 10^{-8} para forças precisas.
ISMEAR	0	Esquema de smearing. 0 = Gaussiano (semicondutores/isolantes). 1 ou 2 = Methfessel-Paxton (metais). -5 = tetraedros com correções de Blöchl (DOS preciso).
SIGMA	0.05	Largura do smearing em eV. Pequena (0.01–0.05) para semicondutores; maior (0.1–0.2) para metais.
LREAL	.FALSE.	Projeção dos PAW. .FALSE. = espaço recíproco (preciso, células pequenas). Auto = real space (rápido, células grandes >20 átomos).
ALGO	Normal	Algoritmo SCF. Normal = Davidson (robusto). Fast = mistura RMM-DIIS (rápido). All = exato (caro mas estável).

Tag	Valor no exemplo	Significado
NSW	0	Número máximo de passos iônicos. 0 = apenas SCF (sem mover átomos). >0 = otimização/dinâmica.
LWAVE	.FALSE.	Salvar WAVECAR? .FALSE. economiza muito espaço em disco quando não precisamos da função de onda.
LCHARG	.FALSE.	Salvar CHGCAR? .FALSE. para SCFs simples; .TRUE. quando o resultado vai alimentar outro cálculo (banda, DOS).

⚠ Atenção: ISMEAR = 0 (Gaussiano) é correto para semicondutores e isolantes. Para metais, usem ISMEAR = 1 ou 2 (Methfessel-Paxton).

4. O arquivo POSCAR

O POSCAR contém toda a informação geométrica: vetores da célula unitária, espécies atômicas e suas posições. O formato é rígido — cada linha tem um significado fixo, então sigam o template exatamente. Criem o arquivo POSCAR:

```
Si diamante - celula primitiva
5.430000000000000
0.5000000000000000 0.5000000000000000 0.0000000000000000
0.0000000000000000 0.5000000000000000 0.5000000000000000
0.5000000000000000 0.0000000000000000 0.5000000000000000
Si
2
Direct
0.0000000000000000 0.0000000000000000 0.0000000000000000
0.2500000000000000 0.2500000000000000 0.2500000000000000
```

Linha por linha, o que cada uma significa:

- **Linha 1:** comentário livre. Use para descrever o sistema (não é lido pelo VASP).
- **Linha 2:** fator de escala universal aplicado aos vetores de rede. Aqui usamos 5.43 Å, o parâmetro experimental do silício. Se quisessem escalar a célula em 1 %, bastaria mudar este número para 5.4843 — os vetores das próximas três linhas seriam multiplicados automaticamente.
- **Linhas 3–5:** os três vetores da célula primitiva, em unidades do fator de escala. Para a estrutura FCC os vetores primitivos canônicos são $(\frac{1}{2}, \frac{1}{2}, 0)$, $(0, \frac{1}{2}, \frac{1}{2})$ e $(\frac{1}{2}, 0, \frac{1}{2})$.
- **Linha 6:** rótulos dos elementos. Aqui só temos Si. A ordem importa: tem que bater com a ordem do POTCAR.
- **Linha 7:** número de átomos de cada espécie. Como temos só silício, é apenas '2'. Se tivéssemos Si e O, seria por exemplo '4 8'.
- **Linha 8:** modo de coordenadas. 'Direct' significa coordenadas fracionárias (frações dos vetores de rede). Alternativamente, 'Cartesian' usa coordenadas cartesianas em Å.
- **Linhas 9 em diante:** as posições atômicas. Duas linhas no nosso caso, uma por átomo de Si.

💡 Dica: Coordenadas Direct (fracionárias) são quase sempre preferíveis. Quando vocês escalam a célula, as posições fracionárias permanecem invariantes — perfeito para varreduras de parâmetro de rede. Com coordenadas cartesianas, vocês teriam que escalar manualmente cada posição.

⚠ Atenção: A ordem dos elementos na linha 6 deve ser exatamente a mesma com que vocês vão concatenar os pseudopotenciais para formar o POTCAR. Se trocarem, o VASP usará o POTCAR errado

para cada átomo sem reclamar — o cálculo rodará, mas os resultados serão fisicamente sem sentido. Esse é o erro mais comum (e perigoso) em cálculos VASP.

5. O arquivo KPOINTS

O KPOINTS define como amostramos a zona de Brillouin. Para sistemas bulk, o esquema mais comum é o de Monkhorst-Pack, que gera uma malha regular de pontos-k. Para o silício, usaremos:

```
K-points
0
Gamma
8 8 8
0 0 0
```

As linhas significam:

- **Linha 1:** comentário livre.
- **Linha 2:** número total de pontos-k. 0 significa 'gerar automaticamente' (modo automático).
- **Linha 3:** tipo da malha. 'Gamma' centra a malha em Γ (recomendado para semicondutores e estruturas hexagonais); 'Monkhorst-Pack' (ou apenas 'M') usa o esquema original (recomendado para alguns sistemas cúbicos).
- **Linha 4:** subdivisões em cada direção da rede recíproca. '8 8 8' significa 8 pontos em cada uma das três direções, o que dá $8^3 = 512$ pontos na zona inteira. Por simetria, esses 512 são reduzidos a apenas 60 pontos-k irredutíveis para a estrutura diamante — só os irredutíveis são realmente calculados.
- **Linha 5:** deslocamento da malha. '0 0 0' significa centrar em Γ (sem deslocamento).

 **Dica:** Para qualquer sistema novo, façam um estudo de convergência: rodem o mesmo cálculo com $4 \times 4 \times 4$, $6 \times 6 \times 6$, $8 \times 8 \times 8$, $10 \times 10 \times 10$ e vejam a partir de qual malha a energia total para de mudar. Para o silício, $8 \times 8 \times 8$ já é bem convergido.

 **Atenção:** Para estruturas hexagonais (como grafeno ou h-BN), utilizem sempre 'Gamma' centered, pois 'Monkhorst-Pack' produz uma malha que quebra a simetria hexagonal.

6. O arquivo POTCAR

Diferentemente dos outros três arquivos, vocês não escrevem o POTCAR à mão. Ele é construído concatenando os arquivos POTCAR oficiais que vêm com a instalação do VASP, na mesma ordem em que os elementos aparecem no POSCAR.

No Heisenberg, os pseudopotenciais PAW estão em um diretório do sistema. Após carregar o módulo do VASP, vocês podem localizá-los com:

```
module load vasp/6.2.0
echo $VASP_PP_PATH      # caminho dos pseudopotenciais
ls $VASP_PP_PATH/potpaw_PBE.54/  # pasta com PAW-PBE
```

 **Atenção:** Se a variável VASP_PP_PATH não estiver definida, perguntem ao suporte do cluster o caminho correto. Normalmente é algo como `/opt/vasp/pseudopotentials/potpaw_PBE.54/`.

Para o nosso silício, só precisamos do POTCAR do Si. Construam o arquivo POTCAR com:

```
cp $VASP_PP_PATH/potpaw_PBE.54/Si/POTCAR ./POTCAR
```

Se tivessem múltiplos elementos, por exemplo Si e O na ordem do POSCAR, fariam:

```
cat $VASP_PP_PATH/potpaw_PBE.54/Si/POTCAR \
    $VASP_PP_PATH/potpaw_PBE.54/O/POTCAR > POTCAR
```

Verifiquem se o POTCAR foi construído corretamente examinando as linhas TITEL:

```
grep TITEL POTCAR
```

A saída deve mostrar uma linha por elemento, na ordem correta. Para Si:

```
TITEL = PAW_PBE Si 05Jan2001
```

 **Atenção:** Existem múltiplos POTCARs para alguns elementos (ex: para o silício temos Si, Si_GW, Si_h, Si_sv). Para cálculos padrão de estado fundamental, sempre usem o POTCAR sem sufixo, que está bem testado e tem o conjunto de elétrons de valência apropriado. Os outros são para casos especiais (cálculos GW, alta pressão, semi-core).

7. O script SLURM

Como nas aulas de ORCA, no Heisenberg submetemos cálculos via SLURM. O script para o VASP tem algumas diferenças importantes: usamos a partição AMD do cluster, mpirun em vez de execução serial, e exportamos OMP_NUM_THREADS=1 para evitar conflito entre OpenMP e MPI.

Criem o arquivo run_vasp.sh com o seguinte conteúdo:

```
#!/bin/bash
#SBATCH -n 16
#SBATCH --ntasks-per-node=16
#SBATCH -p cpu
#SBATCH --mem=32000mb

module load vasp/6.2.0

hostname
export OMP_NUM_THREADS=1
mpirun -n 16 vasp_std > vasp.out
```

Linha por linha:

- **#SBATCH -n 16:** solicita 16 tarefas MPI no total.
- **#SBATCH --ntasks-per-node=16:** todas as 16 tarefas em um único nó (mais eficiente para sistemas pequenos: evita comunicação entre nós).
- **#SBATCH -p cpu:** submete à partição com os nós AMD do Heisenberg, onde o VASP 6.2.0 foi compilado e otimizado.
- **#SBATCH --mem=32000mb:** solicita 32 GB de memória. Generoso para o silício bulk, mas seguro como padrão.
- **module load vasp/6.2.0:** carrega o ambiente do VASP, definindo PATH, bibliotecas MPI e variáveis como VASP_PP_PATH.
- **export OMP_NUM_THREADS=1:** garante que cada processo MPI use apenas um thread, evitando oversubscription. Se vocês não setarem isso, o desempenho pode cair drasticamente.
- **mpirun -n 16 vasp_std > vasp.out:** executa o VASP em paralelo com 16 processos. A saída padrão vai para vasp.out.

 **Atenção:** Existem três binários do VASP: vasp_std (padrão, malhas k gerais), vasp_gam (otimizado para cálculo só no ponto Γ , célula grande), vasp_ncl (não-colinear/spin-órbita). Para a maioria dos cálculos, vocês utilizarão vasp_std, como no exemplo acima.

 **Dica:** Para o silício bulk com célula tão pequena, 16 cores são mais do que suficiente, talvez até demais. Lembrem-se de que paralelismo excessivo em sistemas pequenos é contraproducente: o overhead de comunicação supera o ganho. Quer ganhar vários pontos extra? Mostre como escala a

performance do VASP para o silício em função do número de cores. Após terminar venha até a minha mesa e me mostre o resultado.

8. Submetendo o cálculo

Com os cinco arquivos no diretório (INCAR, POSCAR, KPOINTS, POTCAR, run_vasp.sh), confira tudo:

```
ls
# Deve mostrar:
#  INCAR  POSCAR  KPOINTS  POTCAR  run_vasp.sh
```

Submetam o cálculo:

```
sbatch run_vasp.sh
```

O SLURM responderá com o número do job, algo como Submitted batch job 12345. Acompanhem o progresso com:

```
squeue -u $USER          # ver o estado do job
tail -f vasp.out         # acompanhar a saída em tempo real
```

Para um silício bulk com 8×8×8 pontos-k, o cálculo deve terminar em poucos segundos a um minuto. Para sistemas maiores, podem levar horas ou dias.

9. Analisando os resultados

Quando o cálculo terminar, vocês terão uma série de arquivos de saída. Os mais importantes são:

Arquivo	Conteúdo e quando usar
OUTCAR	O log completo do cálculo: todas as iterações SCF, forças, tensor de tensões, autovalores, ocupações, parâmetros utilizados. É o arquivo mais importante para depuração.
OSZICAR	Resumo conciso da convergência SCF: uma linha por iteração com energia e variação. Ideal para verificar rapidamente se o cálculo convergiu.
CONTCAR	Geometria final no mesmo formato do POSCAR. Em relaxações, é o que vocês utilizam como ponto de partida do próximo cálculo: cp CONTCAR POSCAR.
vasprun.xml	Arquivo XML estruturado contendo praticamente todos os resultados. É o que ferramentas como pymatgen, ASE e py4vasp leem para análises automatizadas.
EIGENVAL	Autovalores de Kohn-Sham (energias das bandas) em cada ponto-k. Base para gráficos de estrutura de bandas.
DOSCAR	Densidade de estados (total e, se LORBIT estiver ativado, projetada em átomos/orbitais).
CHGCAR	Densidade de carga em uma grade 3D. Pode ser visualizado com VESTA ou VMD. Gerado apenas se LCHARG = .TRUE.
WAVECAR	Função de onda. Arquivo enorme — gere apenas quando for reaproveitar o cálculo (LWAVE = .TRUE.).
vasp.out	Saída padrão (stdout) redirecionada pelo script SLURM. Mostra o progresso do cálculo em tempo real.

9.1. Verificar convergência

O primeiro passo é sempre verificar se o SCF convergiu. Olhem o final do OSZICAR:

```
tail -20 OSZICAR
```

Vocês verão algo como:

	N	E	dE	d eps	ncg	rms	rms(c)
DAV:	1	0.142533856947E+02	0.14253E+02	-0.21847E+03	3072	0.563E+02	
DAV:	2	-0.114789573289E+02	-0.25732E+02	-0.24315E+02	3344	0.530E+01	
DAV:	3	-0.115893824537E+02	-0.11043E+00	-0.11043E+00	3024	0.398E+00	
...							
DAV:	8	-0.115921074329E+02	-0.27250E-07	-0.27249E-07	3008	0.124E-03	
1 F= -.11592107E+02 E0= -.11592107E+02 d E =-.115921E+02							

As linhas DAV mostram cada iteração SCF. Quando dE cai abaixo do EDIFF (10^{-6} eV no nosso INCAR), o cálculo converge. A linha final que começa com '1 F=' dá a energia total convergida: aqui, aproximadamente -11,592 eV para a célula primitiva (dois átomos).

Dividindo por 2, temos a energia por átomo: -5,796 eV/átomo. Esse valor depende do POTCAR e funcional usados; é uma referência arbitrária, mas diferenças de energia entre cálculos com o mesmo setup são fisicamente significativas.

9.2. Verificar os parâmetros usados

Sempre confirmam, no início do OUTCAR, quais foram os parâmetros efetivamente usados:

```
grep -E 'ENCUT|ISMEAR|SIGMA|EDIFF|ENMAX' OUTCAR | head
```

Isso é importante porque o VASP pode silenciosamente utilizar valores padrão se vocês esquecerem de definir alguma tag. Conferir é um hábito de higiene.

9.3. Energia, forças e tensões

Para extrair a energia final do OUTCAR:

```
grep 'free energy TOTEN' OUTCAR | tail -1
```

Para ver o tensor de tensões na célula (útil para saber se a estrutura está em equilíbrio):

```
grep -A 1 'in kB' OUTCAR | tail -2
```

Se as componentes diagonais do tensor forem todas próximas de zero (digamos, menos de 1 kbar), a célula está perto do equilíbrio. Para o silício a 5,43 Å (parâmetro experimental, à temperatura ambiente), a PBE prevê uma pressão pequena mas não-nula — o PBE superestima o parâmetro de rede em cerca de 1 %.

10. Exercícios e próximos passos

Como exercício, façam o seguinte:

1. Convergência de ENCUT: rodem o mesmo cálculo com ENCUT = 250, 300, 350, 400, 450, 500 eV e plotem a energia total versus ENCUT. A partir de qual valor a energia muda menos de 1 meV/átomo?
2. Convergência de pontos-k: fixem ENCUT no valor convergido do item 1 e variem a malha (4×4×4, 6×6×6, 8×8×8, 10×10×10, 12×12×12). Quando a energia para de mudar?
3. Otimização do parâmetro de rede: variem o fator de escala do POSCAR de 5.30 a 5.55 em passos de 0.05 e ajustem a curva E(a) a um polinômio de segundo grau (ou Birch-Murnaghan). Qual o parâmetro de rede de equilíbrio previsto pelo PBE? Como ele se compara ao experimental (5.431 Å)?

4. Otimização automática: alterem o INCAR para fazer uma relaxação completa da célula (ISIF = 3, IBRION = 2, NSW = 30, EDIFFG = -0.01) e comparem o parâmetro otimizado com o que vocês obtiveram manualmente no item 3. Eles devem coincidir.
5. Estrutura de bandas e DOS: após relaxar a estrutura, façam um SCF salvando CHGCAR (LCHARG=TRUE.), e em seguida rodem um cálculo não-SCF (ICHARG=11) ao longo de um caminho L- Γ -X-W-K- Γ na zona de Brillouin para obter a estrutura de bandas do silício. Comparem o gap calculado com o experimental (~ 1.17 eV) — vão observar o famoso 'gap problem' do DFT-GGA.

11. Resumo

Vocês acabaram de fazer o pulo da química quântica molecular para a estrutura eletrônica de sólidos. Os conceitos físicos não mudaram, mas o ferramental sim: agora pensamos em termos de ondas planas, pontos-k e células periódicas, em vez de gaussianas atômicas e moléculas isoladas.

As dificuldades práticas mais comuns no início são (a) esquecer de bater a ordem dos elementos entre POSCAR e POTCAR, (b) não fazer testes de convergência de ENCUT e pontos-k antes de produzir, (c) usar smearing inadequado para o sistema (Gaussiano em metal ou Methfessel-Paxton em isolante para extrair energia muito precisa). Atenção a esses três pontos resolve 90 % dos problemas que aparecem.

Nas próximas aulas vamos avançar para sistemas com superfícies, defeitos pontuais, propriedades magnéticas e DFT+U. Toda a base que vocês construíram agora será aproveitada.