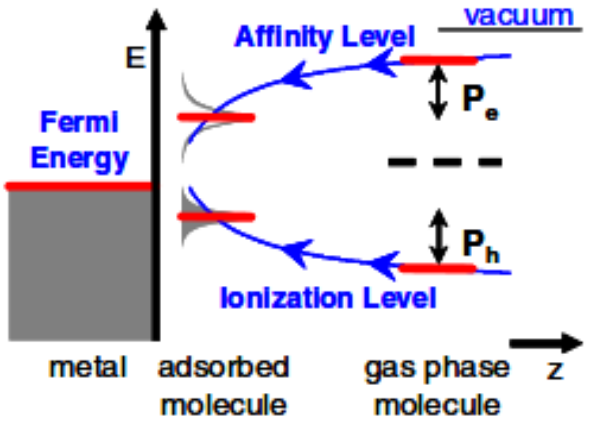
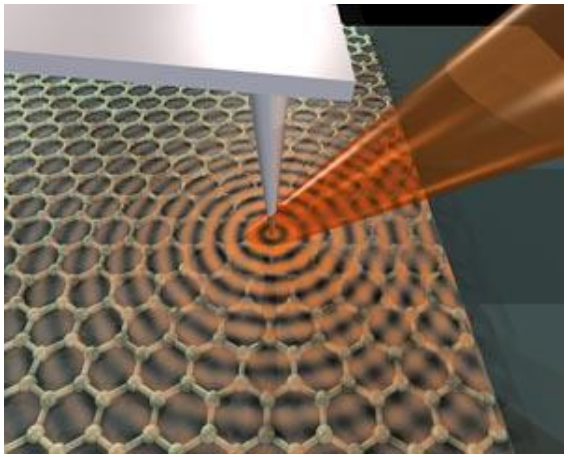
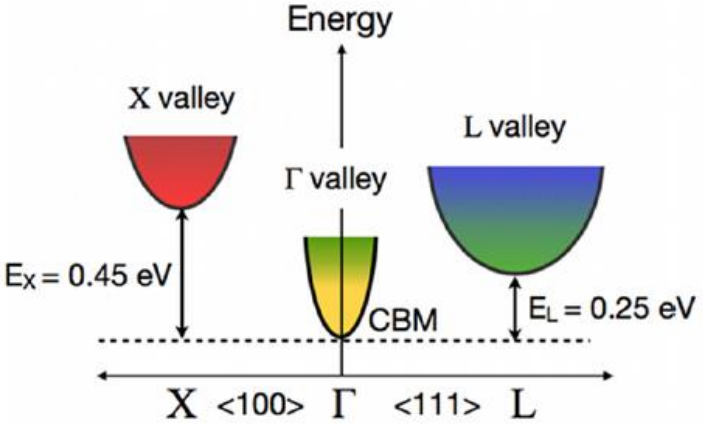
Computation and inversion of the dielectric matrix







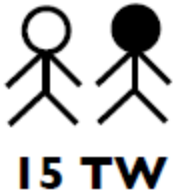
Derek Vigil-Fowler

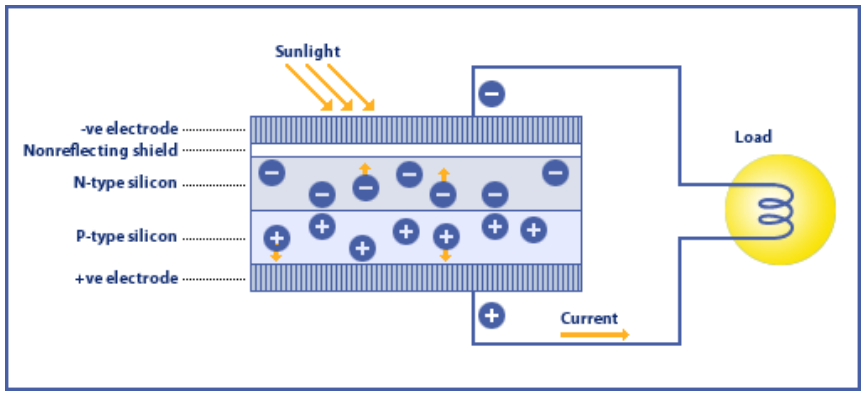
UC-Berkeley and LBNL

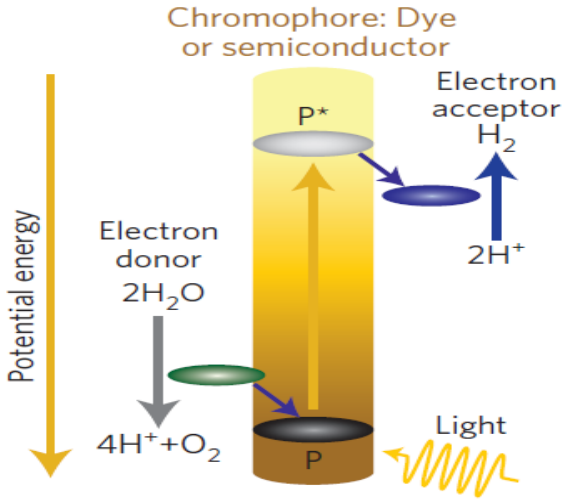
BW Symposium 05/12/15

Email - vigil@berkeley.edu

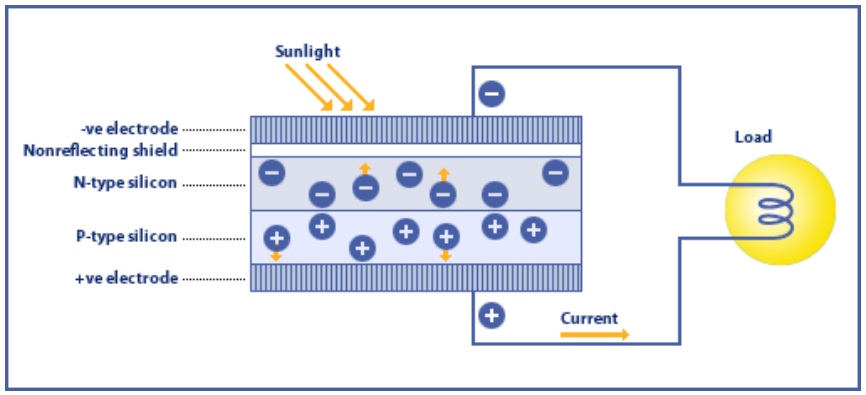
Materials Science for Energy, Technology

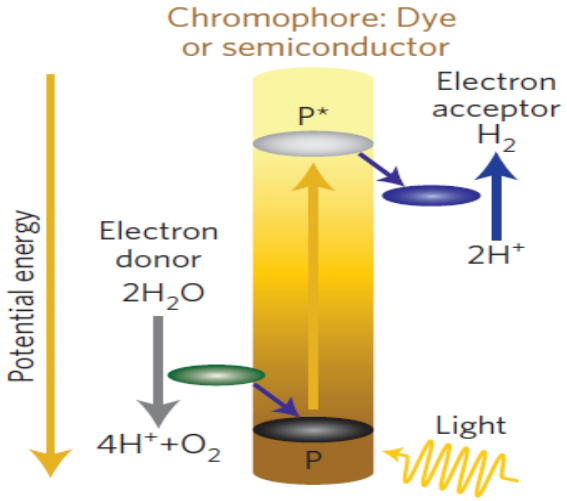


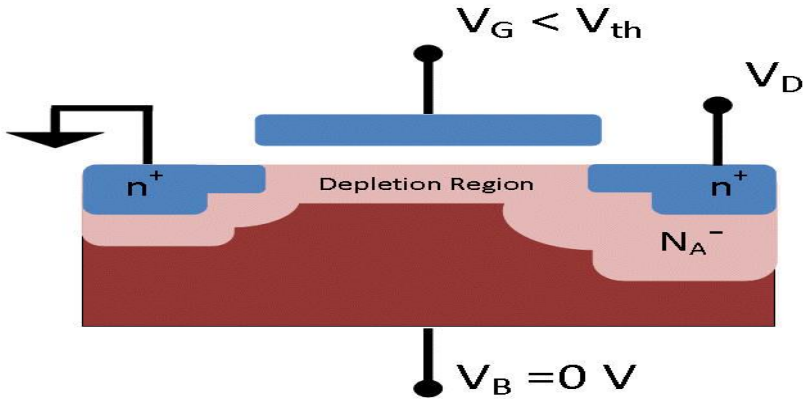




Materials Science for Energy, Technology



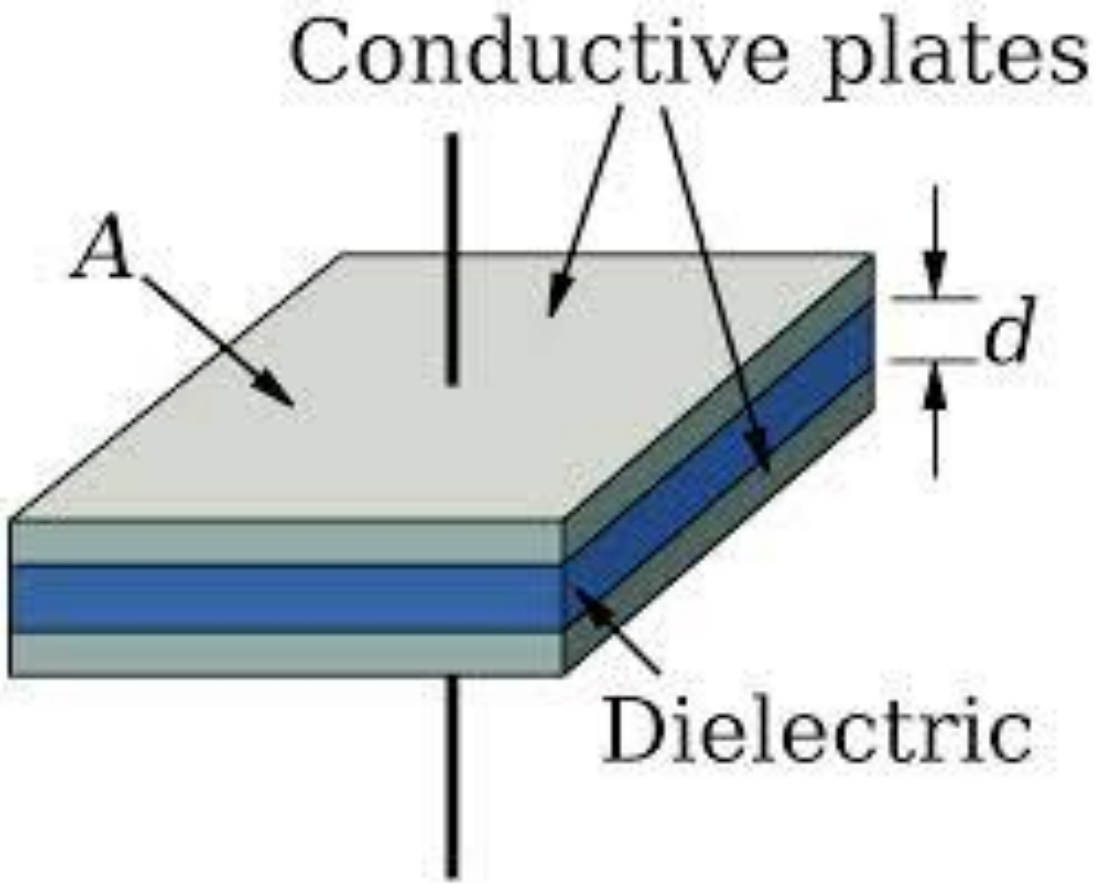




Dielectric response :

Dielectric response : E&M

Dielectric response : E&M



Dielectric response : quantum mechanics

Dielectric response : quantum mechanics

$$\chi_{\mathbf{G}\mathbf{G}'}(\omega) = \sum_{n_v, n_c} M_{\mathbf{G}, n_v n_c} M^*_{n_v n_c, \mathbf{G}'} f_{n_v n_c}(\omega)$$

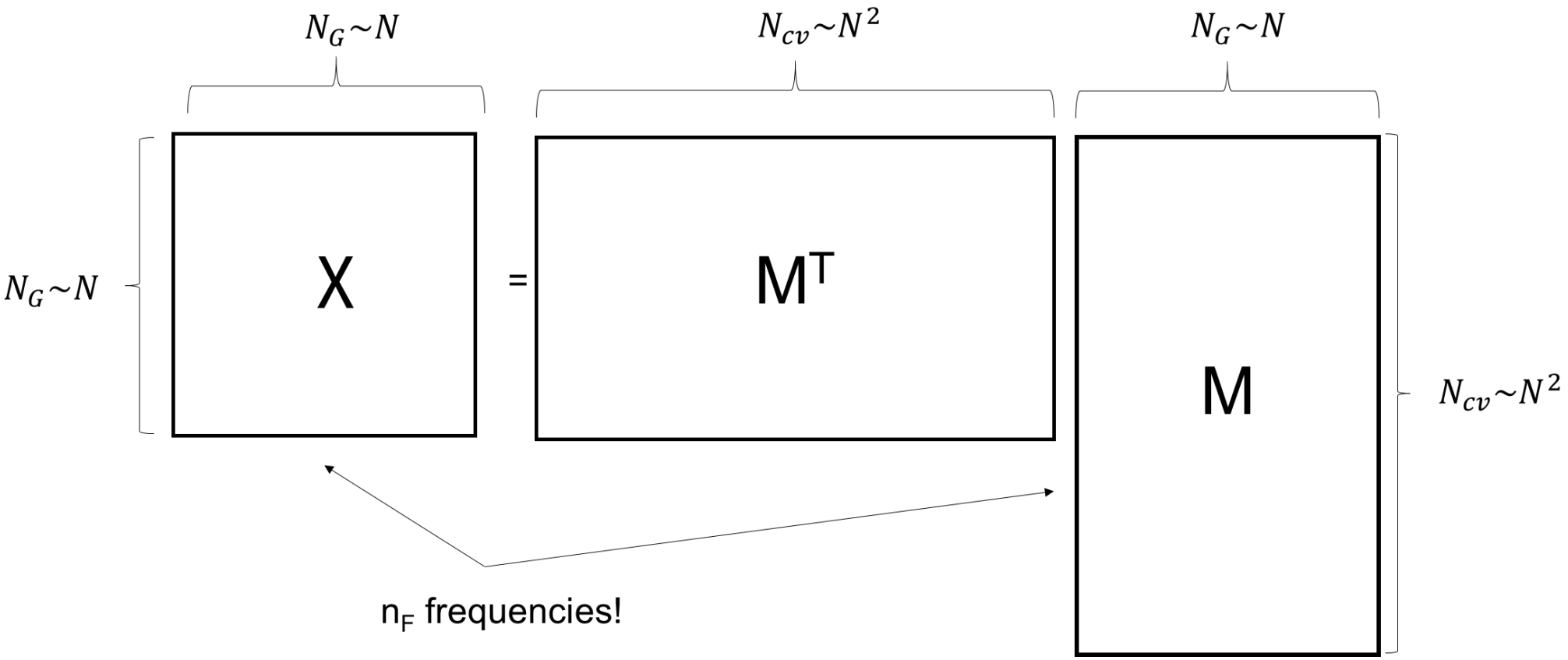
Dielectric response : quantum mechanics

$$\chi_{\mathbf{G}\mathbf{G}'}(\omega) = \sum_{n_v, n_c} M_{\mathbf{G}, n_v n_c} M^*_{n_v n_c, \mathbf{G}'} f_{n_v n_c}(\omega)$$

$$\epsilon_{\mathbf{G}\mathbf{G}'}(\omega) = \delta_{\mathbf{G}\mathbf{G}'} - v(\mathbf{G}) \chi_{\mathbf{G}\mathbf{G}'}(\omega)$$

Pictorially

Pictorially



How to do one big matrix multiplication + inversion?

How to do one big matrix multiplication + inversion?

Parallelism!

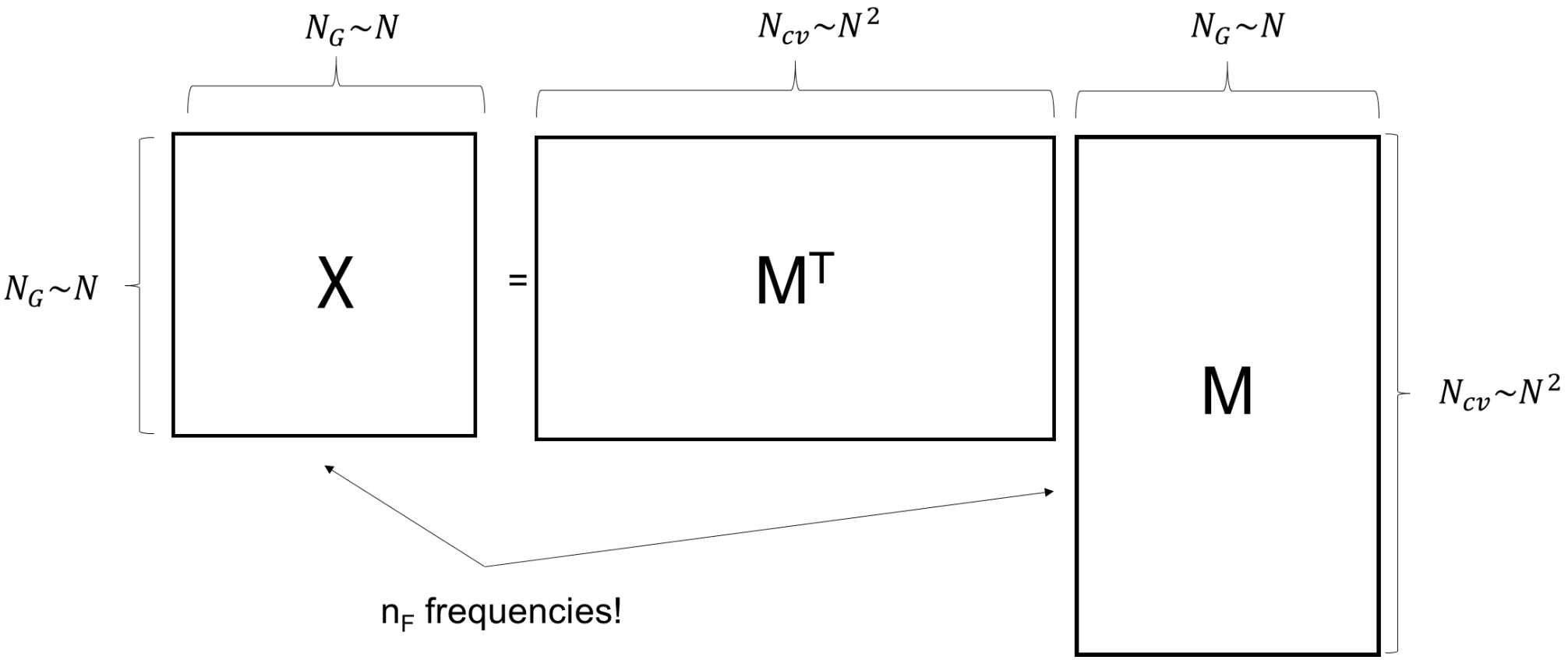
How to do one big matrix multiplication + inversion?

BLAS

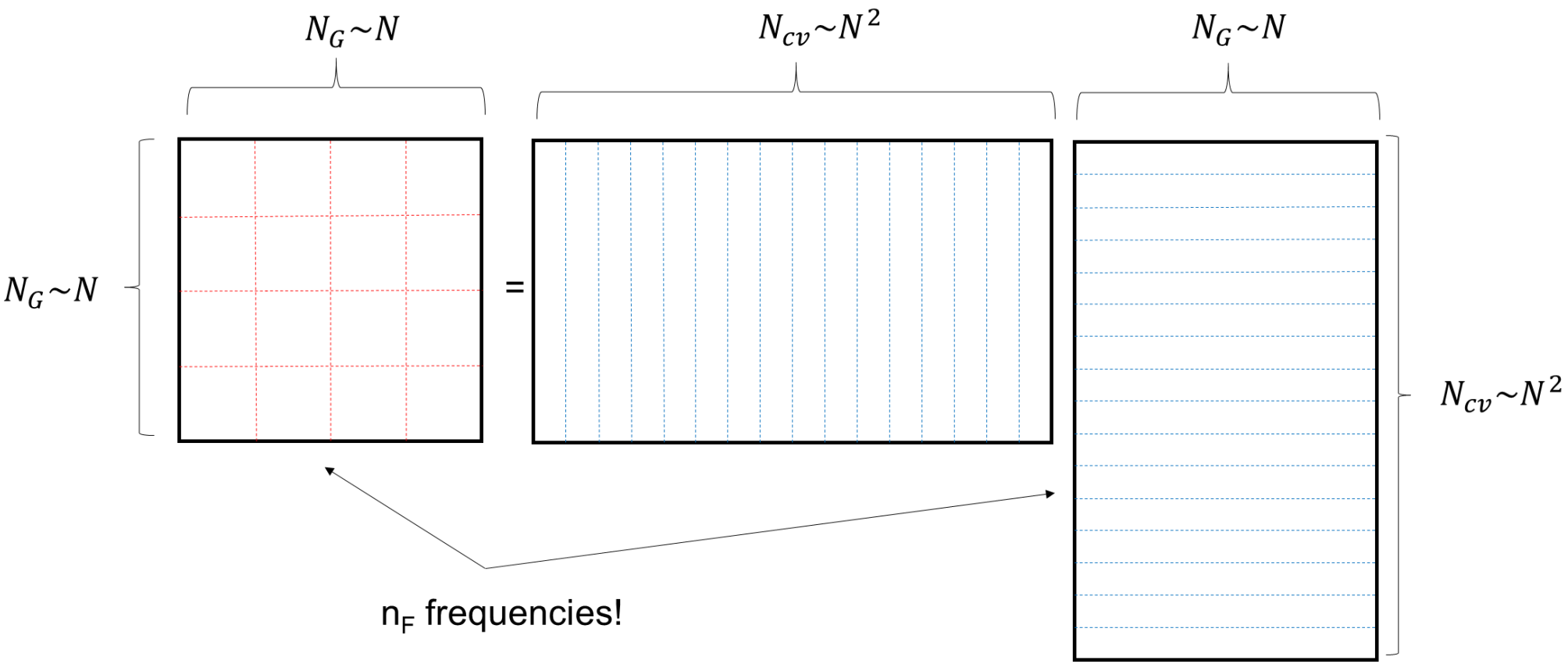
+ ScaLAPACK

+ MPI/OpenMP

Distributed matrix multiplication



Distributed matrix multiplication



Problem with scheme : many frequencies done serially

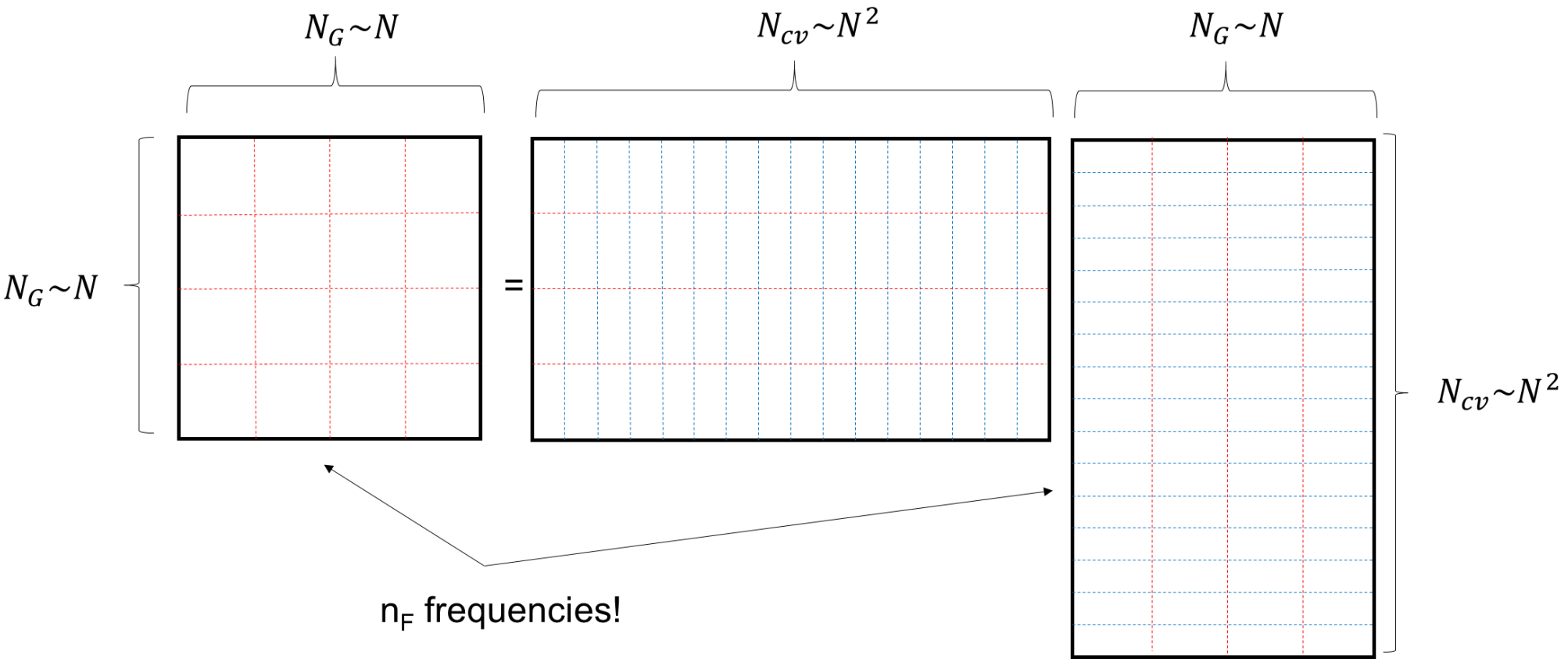
- Lots of communication and array assignments

- All processors work on 1 frequency
  - But ScaLAPACK doesn't scale past ~ few hundred processors!
  - Smaller problems : can't utilize ScaLAPACK enough

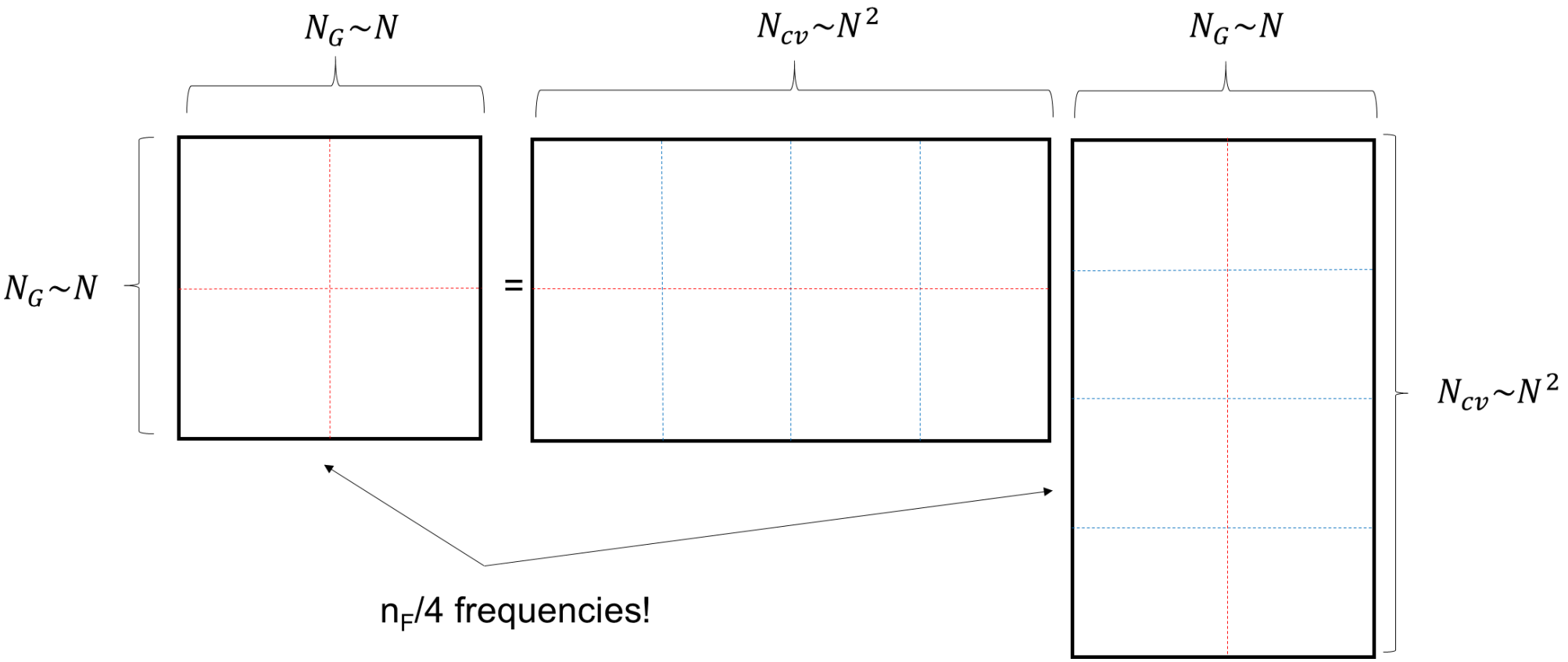
→ Wasted processors

Solution : do many frequencies in parallel!

Solution : do many frequencies in parallel!



Solution : do many frequencies in parallel!



Results

Results

	Bulk Si with 288 proc			CO with 144 proc		
nfreq_par	1	2	8	1	2	8
Matmul total	13.12	8.934	4.395	9.31	6.89	2.13
Matmul prep	10.75	7.08	3.23	1.27	1.01	0.66
Matmul dgemm	2.17	1.75	1.135	1.85	1.60	0.90
Matmul comm	0.2	0.104	0.027	6.18	4.27	0.57
Invert total	0.744	0.26	0.064	5.28	2.60	0.93

Conclusions

- Parallelizing over frequencies reduced communication, array assignment, and saturates ScalaPACK : faster run-time.

- Also, for big problems will allow scaling to higher processors counts for the *frequency-dependent* inverse dielectric matrix, a quantity of wide interest

Acknowledgments

- Blue Waters Graduate Fellowship
- Jack Deslippe – NERSC

- Felipe Homrich da Jornada – UC-Berkeley

This research is part of the Blue Waters sustained-petascale computing project, which is supported by the National Science Foundation (awards OCI-0725070 and ACI-1238993) and the state of Illinois. Blue Waters is a joint effort of the University of Illinois at Urbana-Champaign and its National Center for Supercomputing Applications.