# Ground states of strongly coupled quantum systems of light and matter

Nick Rivera

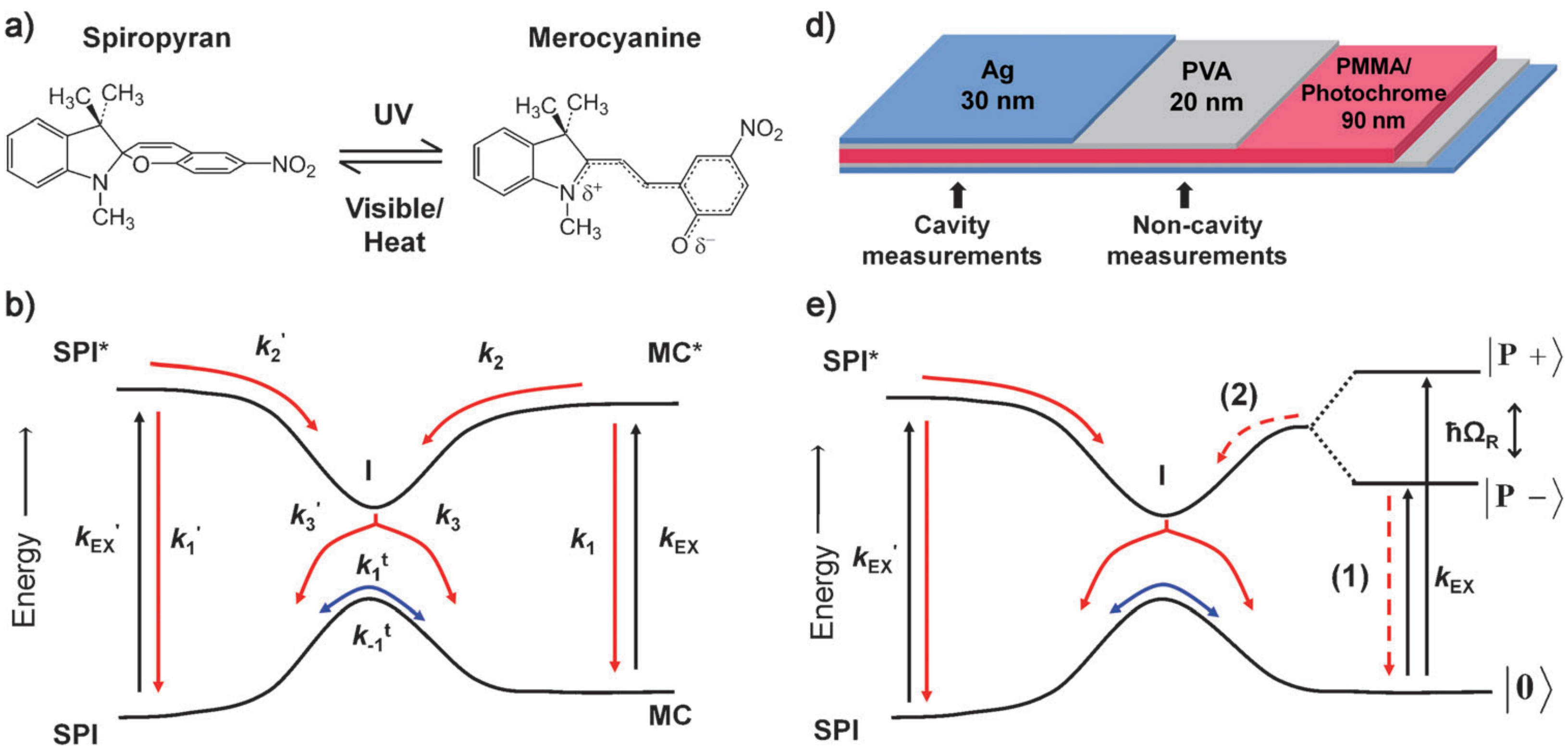
MIT Physics

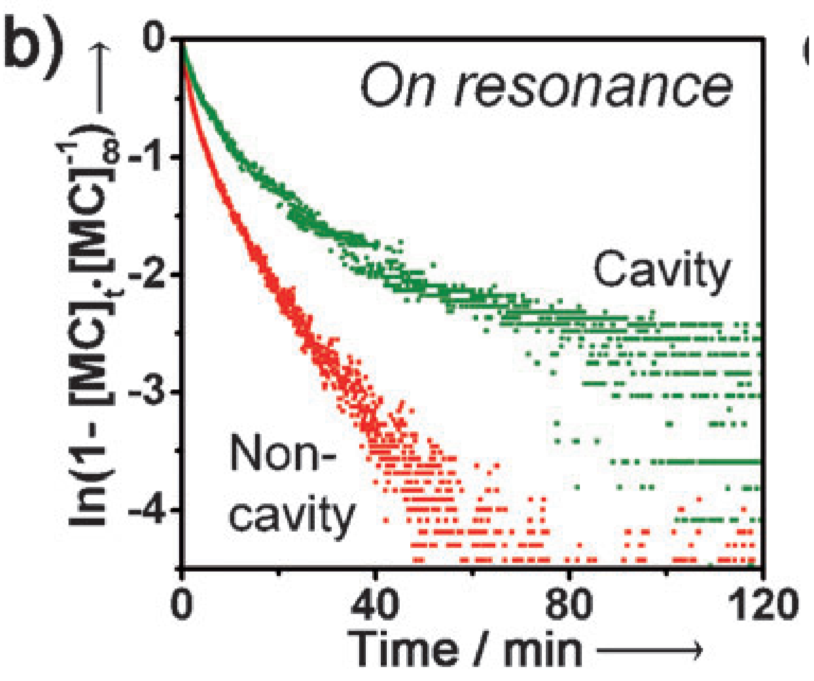
(in collaboration with Dr. Johannes Flick and Prof. Prineha Narang at Harvard)

# The “alchemy” of the quantum vacuum

## Modifying Chemical Landscapes by Coupling to Vacuum Fields**

*James A. Hutchison, Tal Schwartz, Cyriaque Genet, Eloïse Devaux, and Thomas W. Ebbesen**

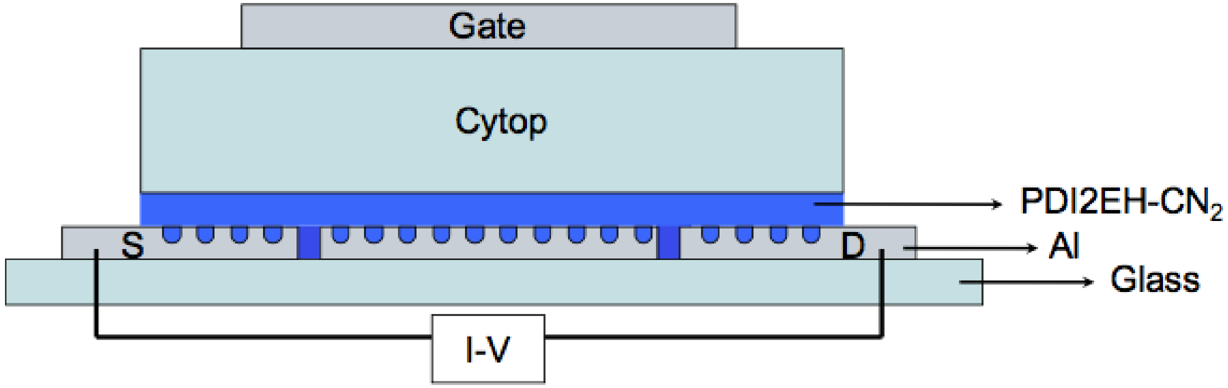


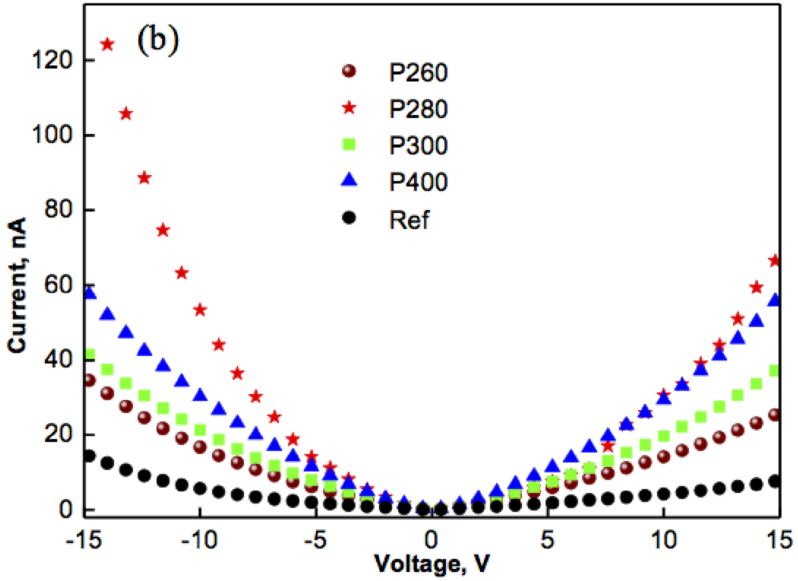


# “Quantum electrodynamical materials science”

## Conductivity in Organic Semiconductors Hybridized with the Vacuum Field

E. Orgiu†,[1] J. George†,[1] J. A. Hutchison†,[1] E. Devaux,[1] J. F. Dayen,[2] B. Doudin,[2] F. Stellacci,[3]
C. Genet,[1] J. Schachenmayer,[4] C. Genes,[5] G. Pupillo,[1,2] P. Samorì,[1] and T. W. Ebbesen*[1]

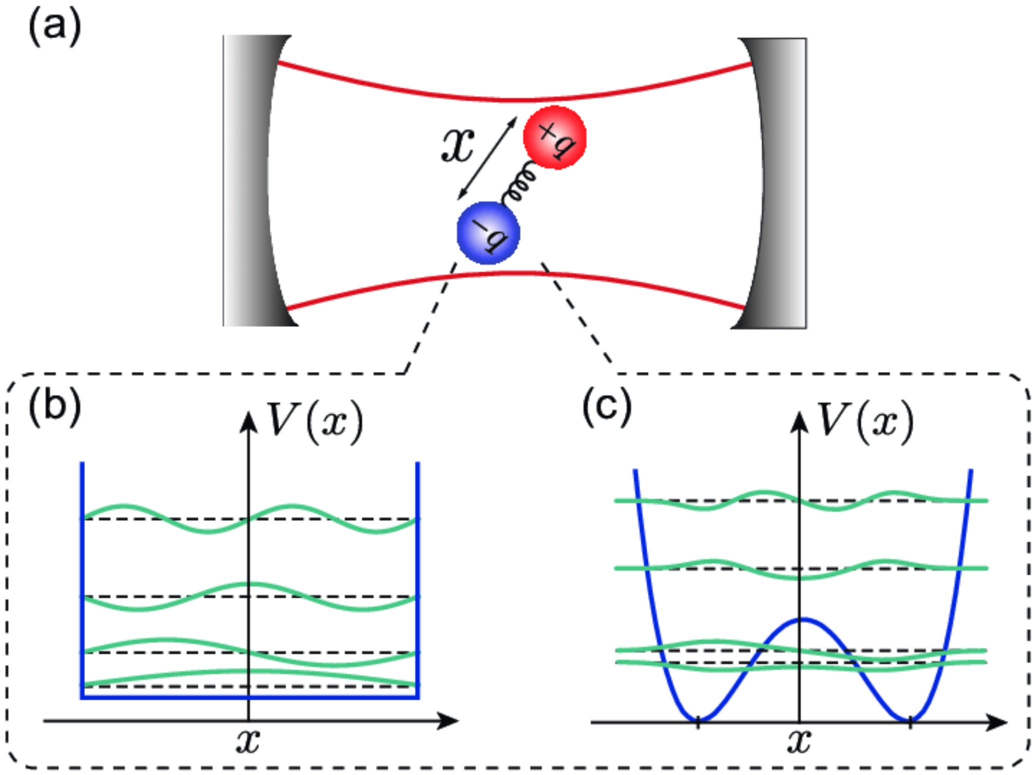


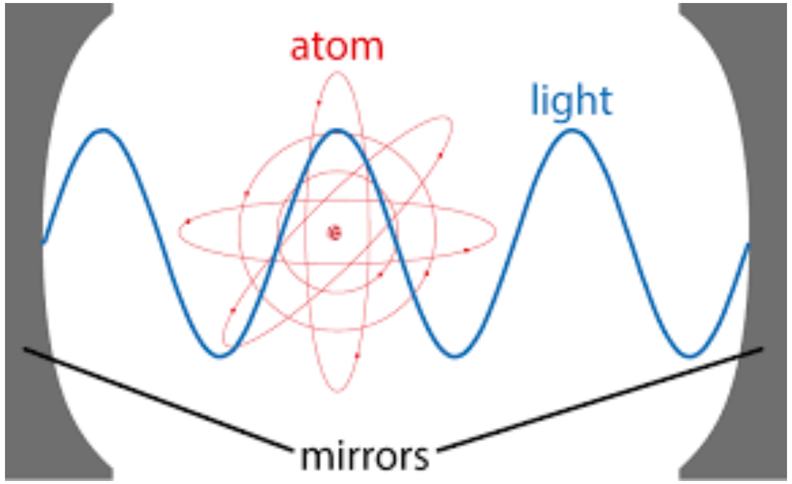


**First principles approaches that capture both the quantum behavior of matter *and* light constitute a new field: *ab initio quantum electrodynamics***

# Simple (but difficult to solve) conceptual example of the system at hand

- A molecule in an optical cavity (electron modes + photon modes)





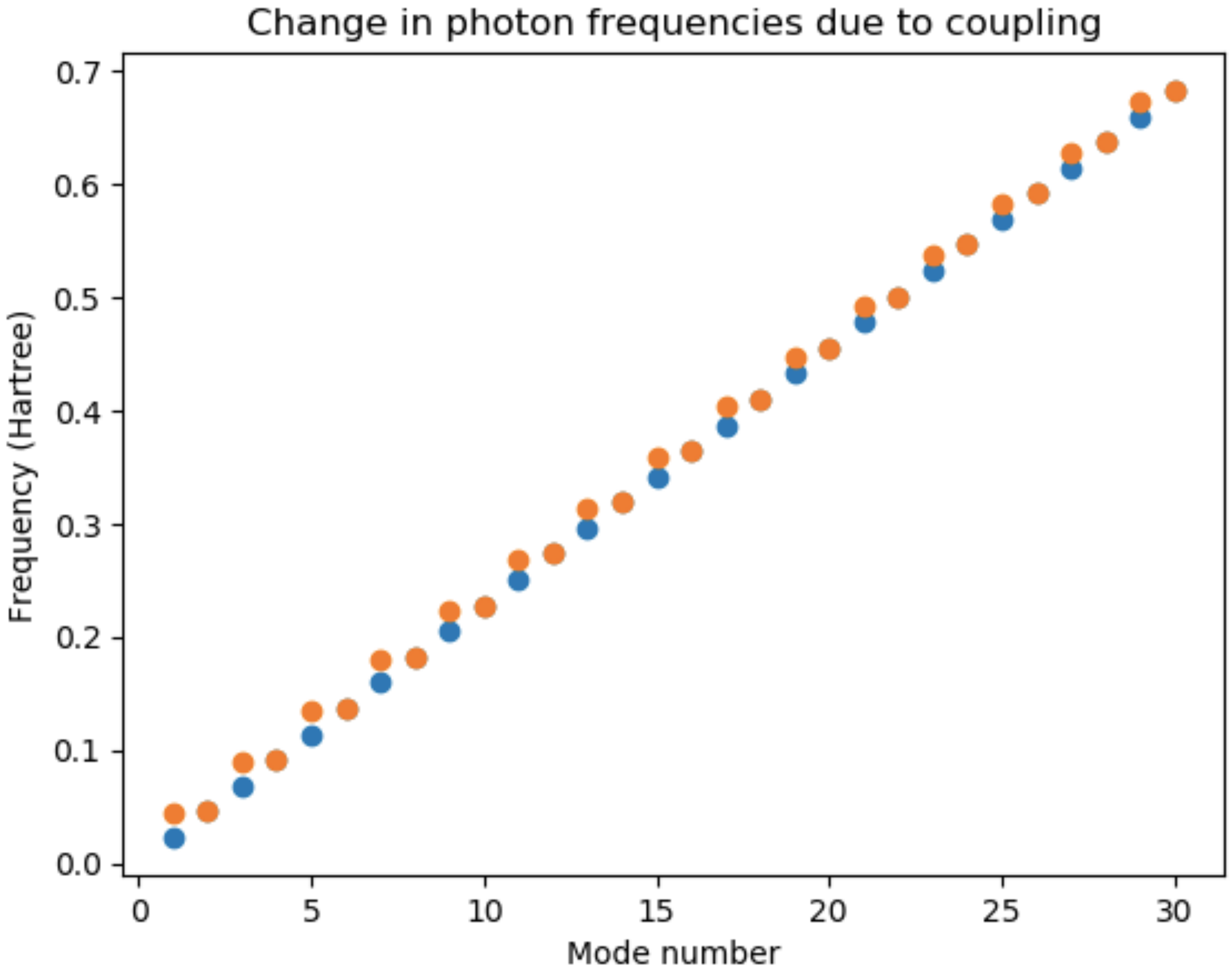
De Bernardis, Jaako, Rabl. Physical Review A. (2018).

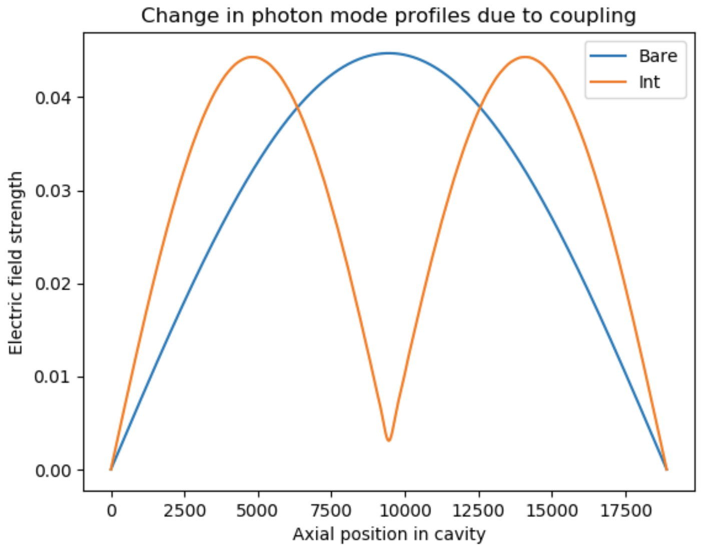
# Overview of routines introduced in this code

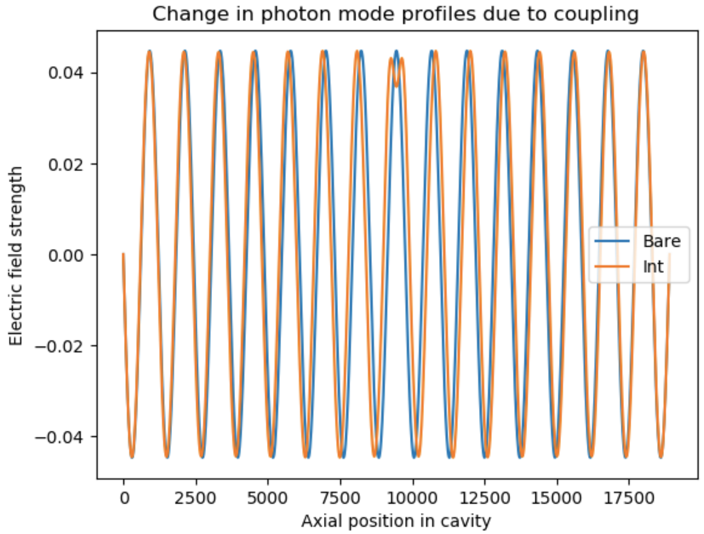
- Methods:
  - Given some user-input electron + photon system, can for example:
    - Find response of electromagnetic field to external probe (time-harmonic dipole)
    - Find energy changes by the electrons induced by the photons
    - Find the behavior of the electron density and electromagnetic field in coupled system
  - **All of these require solving a non-linear eigenproblem or non-linear equation**

- Package dependencies
  - PyPlot, BenchmarkTools, LinearAlgebra, SparseArrays, Arpack, **NLsolve**

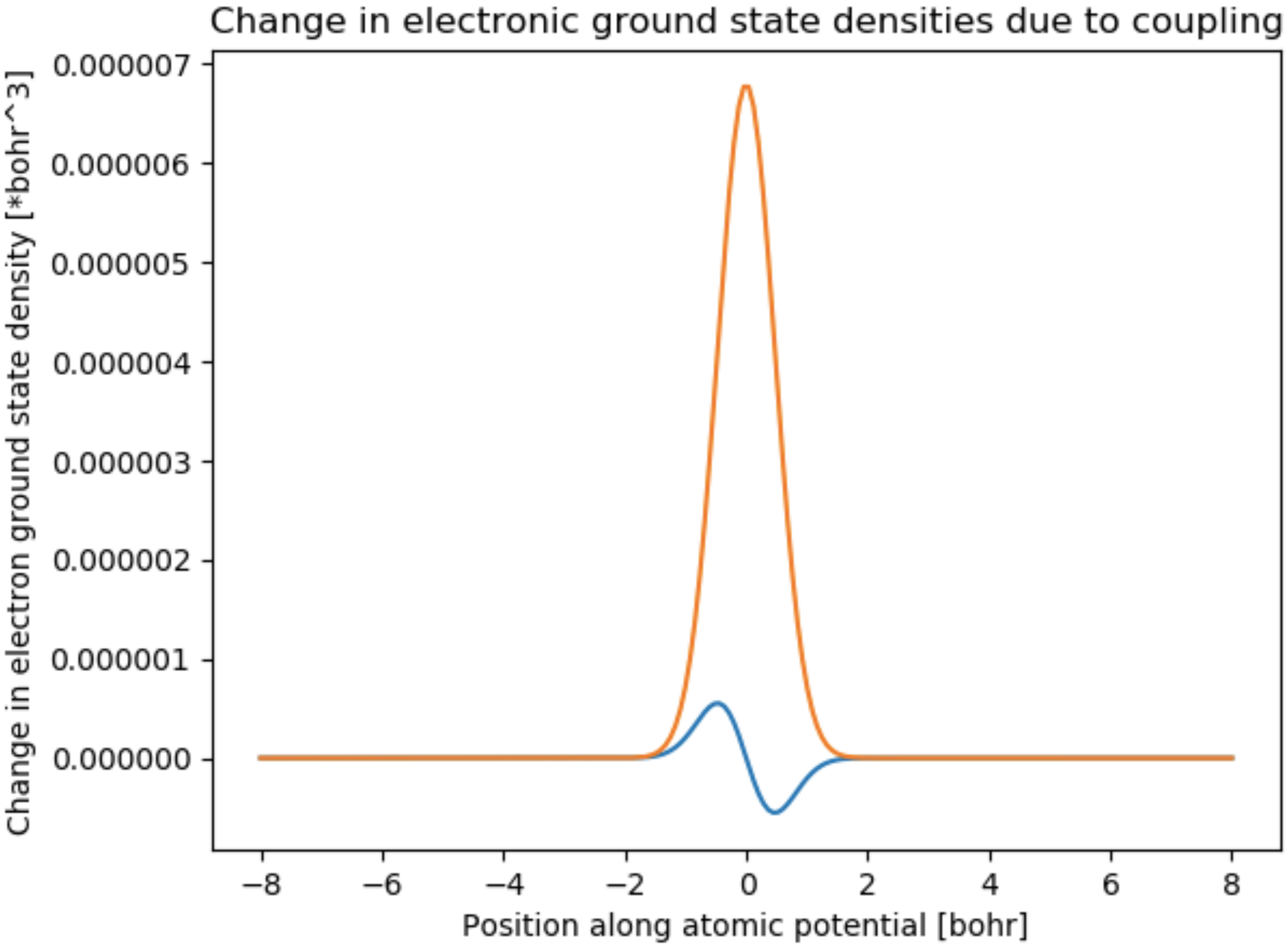
# Results I: profile of the quantum vacuum field created by an atom



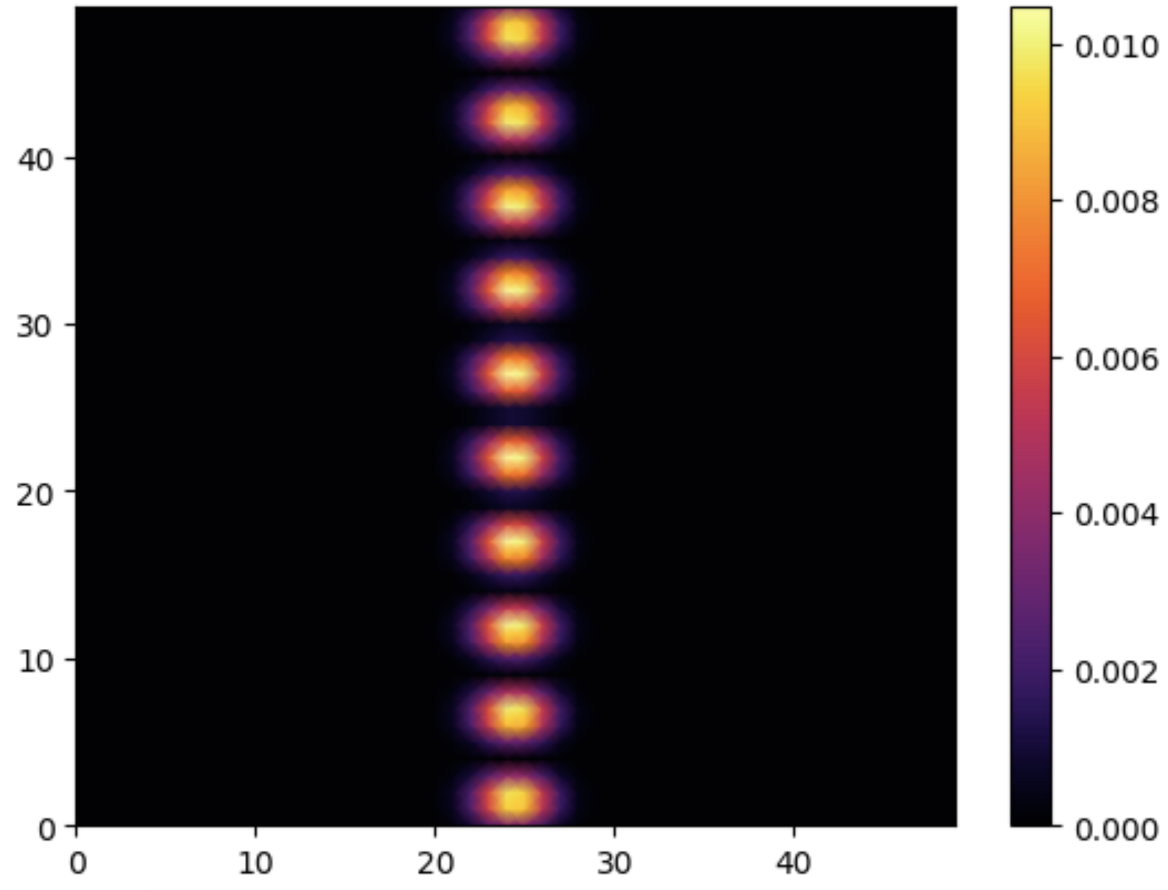


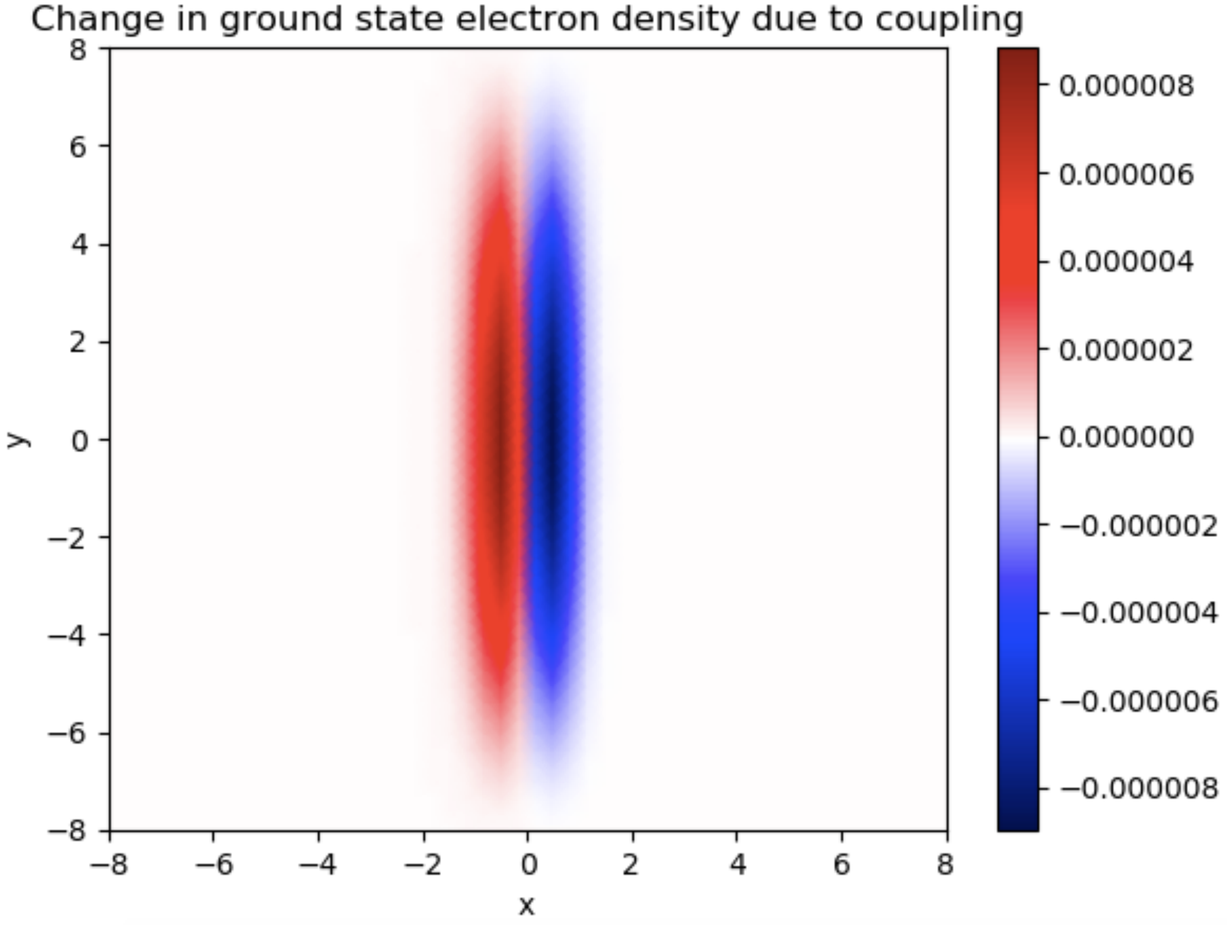


# Results IIa: alteration of the density of electrons by quantum vacuum fields



# Results IIb: alteration of the density of electrons by quantum vacuum fields





# Equations to be solved (an example)

- Simplest level of approximation which would be appropriate when the interaction between light and matter is strong

$$\left[\left(-\frac{1}{2}\nabla^2 + U(\mathbf{r})\right) - q\mathbf{r}\cdot\langle\mathbf{E}(\mathbf{r})\rangle\right]\psi(\mathbf{r}) = E\psi(\mathbf{r})$$

$$\langle\mathbf{E}(\mathbf{r})\rangle = q\langle\mathbf{r}\rangle\sqrt{\frac{2}{\omega_n}}\sum_{n=1}^{N_p}\mathrm{Re}\ \mathbf{F}_n(\mathbf{r})$$

- This equation has a non-local cubic nonlinearity due to the term

$$\psi(\mathbf{r})\int d\mathbf{r}'\ \mathbf{r}'|\psi(\mathbf{r}')|^2$$

# julia NLsolve package is a highly flexible method for solving fixed point problems

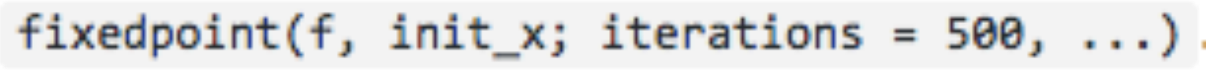
- Often good way to solve a non-linear equation: cast it into the form F(x) = x

$$u'' + au^3 = \lambda u$$

- At kth step: $u_k'' + a u_{k-1}^2 u_k = \lambda_k u_k$ is linear. Solved when

$$f(u_{k-1}) = \mathrm{eigs}(D^2 + a u_{k-1}^2) \approx u_{k-1}$$

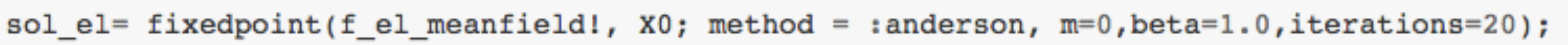
- NLsolve in julia provides a general framework for solving these problems.



```
fixedpoint(f, init_x; iterations = 500, ...)
```

can be an eigensolver (as in the code I wrote)

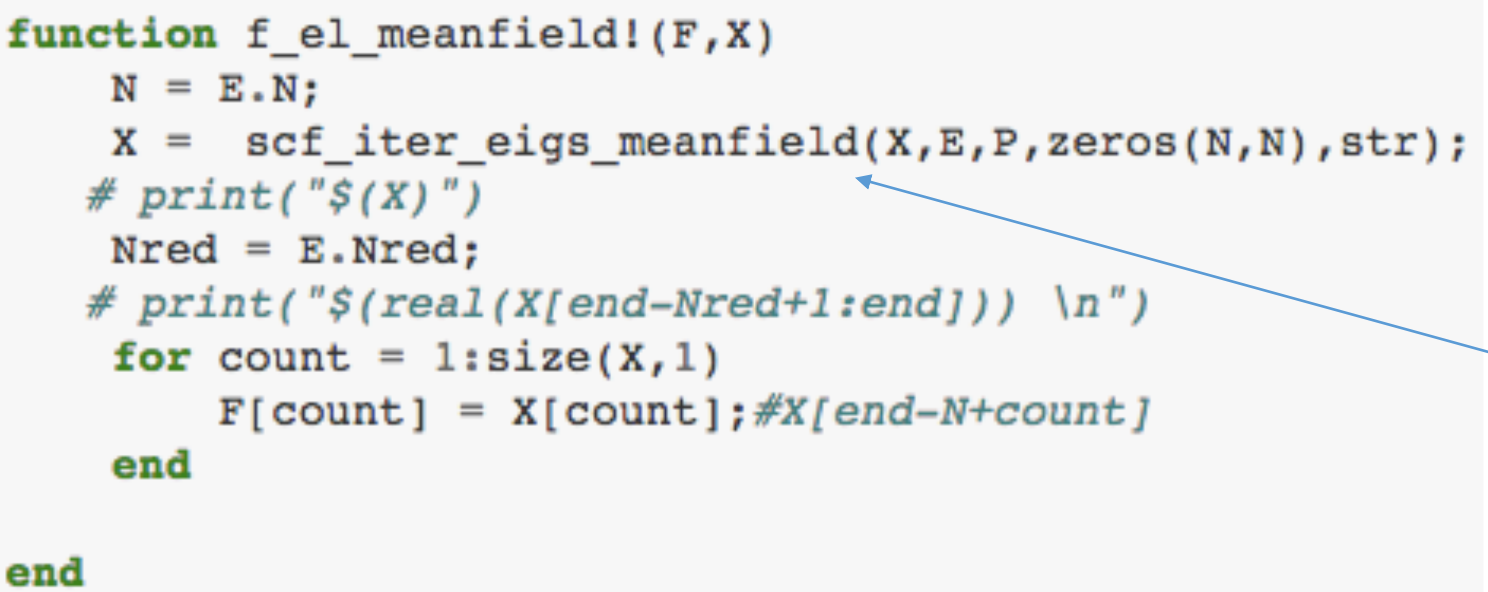
# julia NLsolve package as a highly flexible method for solving fixed point problems



```
sol_el= fixedpoint(f_el_meanfield!, X0; method = :anderson, m=0,beta=1.0,iterations=20);
```

m = 0: Picard iteration

m != 0: Anderson acceleration

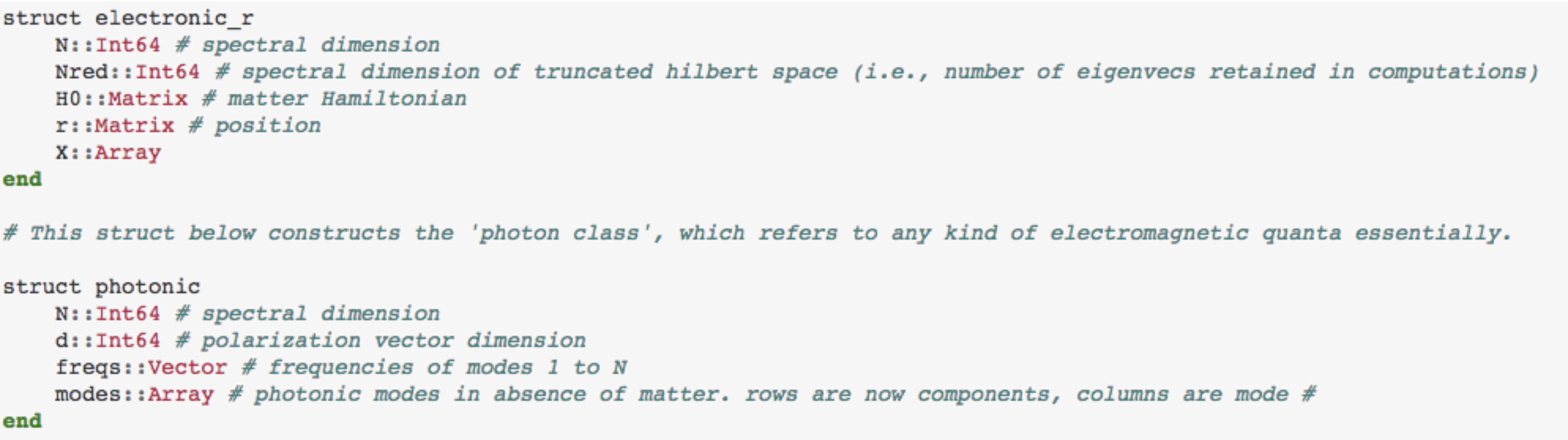


```
function f_el_meanfield!(F,X)
    N = E.N;
    X =  scf_iter_eigs_meanfield(X,E,P,zeros(N,N),str);
   # print("$(X)")
    Nred = E.Nred;
   # print("$(real(X[end-Nred+1:end])) \n")
    for count = 1:size(X,1)
        F[count] = X[count];#X[end-N+count]
    end

end
```

eigensolver which takes photons, constructs effective potential felt by electrons, and diagonalizes

# julia: easily exploit structure through types makes it easy to interface with other software and easy to write common code to different methods



```
struct electronic_r
    N::Int64 # spectral dimension
    Nred::Int64 # spectral dimension of truncated hilbert space (i.e., number of eigenvecs retained in computations)
    H0::Matrix # matter Hamiltonian
    r::Matrix # position
    X::Array
end

# This struct below constructs the 'photon class', which refers to any kind of electromagnetic quanta essentially.

struct photonic
    N::Int64 # spectral dimension
    d::Int64 # polarization vector dimension
    freqs::Vector # frequencies of modes 1 to N
    modes::Array # photonic modes in absence of matter. rows are now components, columns are mode #
end
```

Gives opportunity to “feed in” results from generic quantum eigensolvers and Maxwell eigensolvers like MPB, COMSOL, etc.