

Recent progress in the simulation of time-resolved quantum nanoelectronics

1. Why quantum nanoelectronics at GHz and THz frequencies.
2. A time dependent concept: dynamical control of an interference pattern.
3. Simulating d.c. quantum transport: Kwant
WRITE YOUR HAMILTONIAN AS YOU DO ON THE BLACKBOARD
4. Simulating a.c. quantum transport: T-Kwant
5. Time-resolved quantum transport: mathematical aspects
6. More fancy applications.

Xavier Waintal, SPSMS, INAC CEA Grenoble
With Benoit Gaury, Joseph Weston, Christoph Groth.

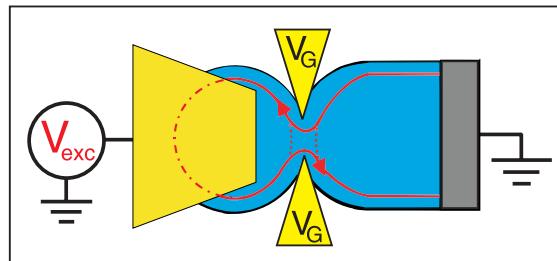
Quantum nanoelectronics in the time domain: why?

- 6 THz = 300K: should room temperature THz electronics ever exist, it must be a fully quantum electronics
- 20 GHz = 1K: time resolved quantum electronics is now possible in the lab. First coherent single electron sources.
- Emergence of new concepts specific to time-dependent physics.

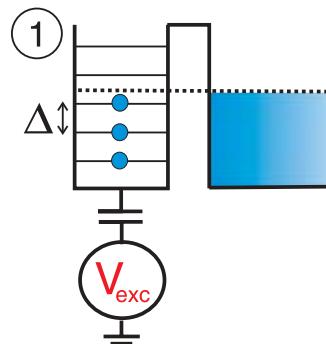
Mesoscopic physics in the time domain.

« Electronic quantum optic » is now coming to the lab

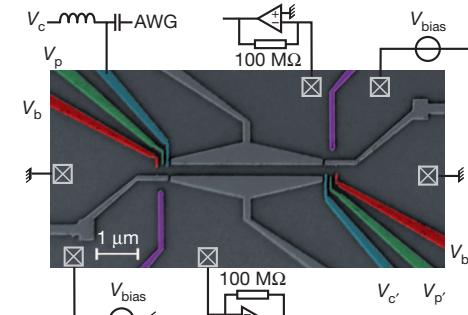
- Single electron source
(Glattli et al. LPA, ENS Paris)



Science 316, 1169 (2007)

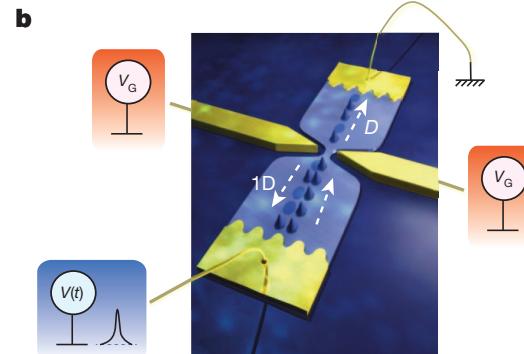


- Single electron source with SAW (Bauerle, Meunier et al Neel, Grenoble, Ritchie et et al, Cambridge)

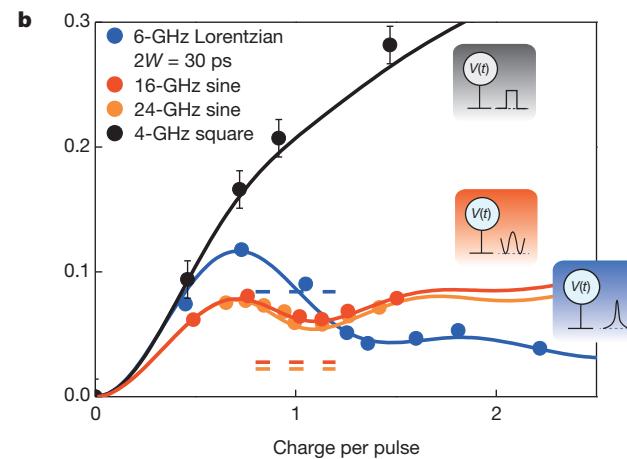


Nature 447, 435 (2011)

- The Leviton: (Glattli et al. SPEC, CEA Saclay)

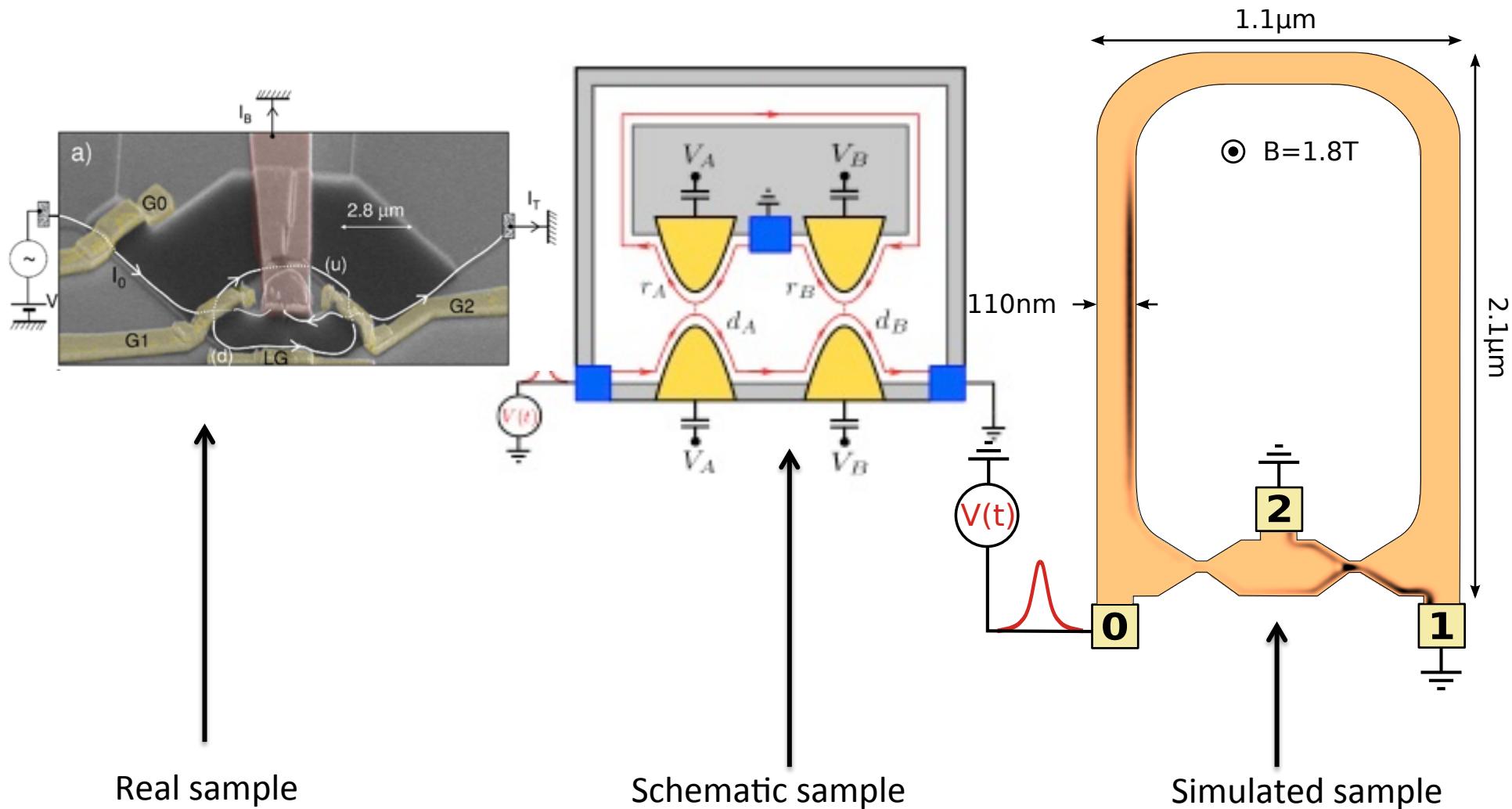


Nature 502, 659 (2013)



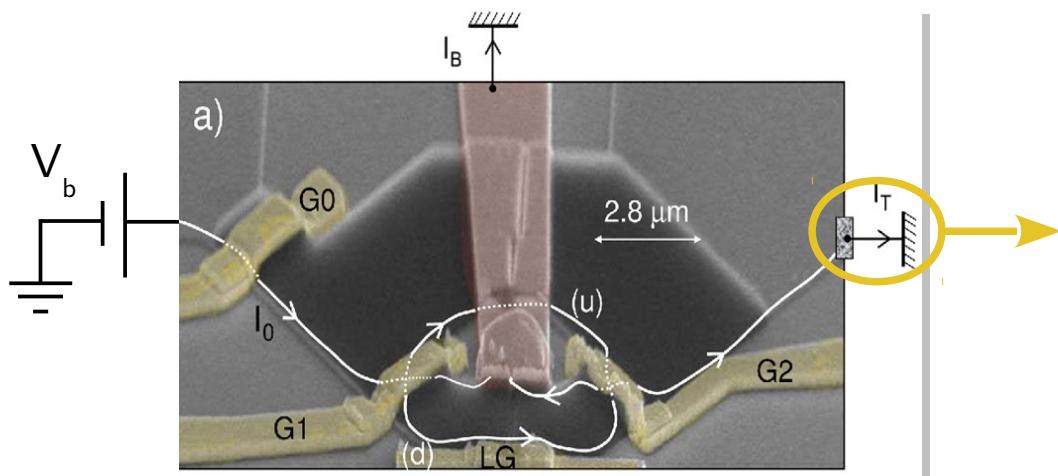
Dynamical control of an interference pattern

Machzender Interferometer



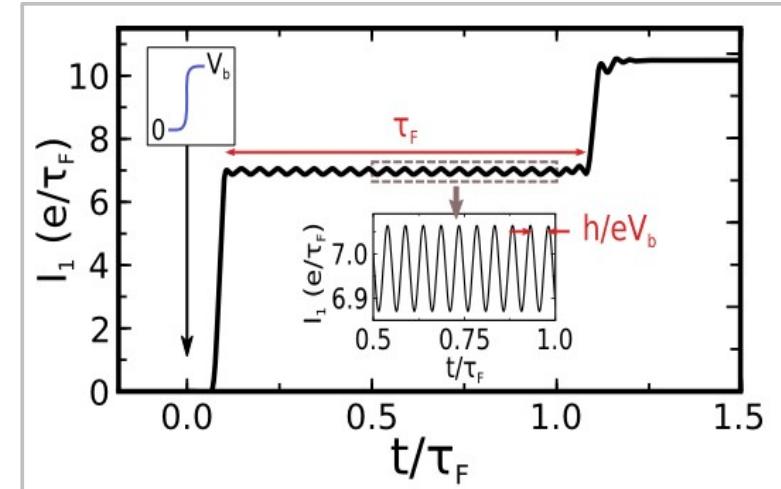
Dynamical control of an interference pattern

Mach-Zehnder interferometer

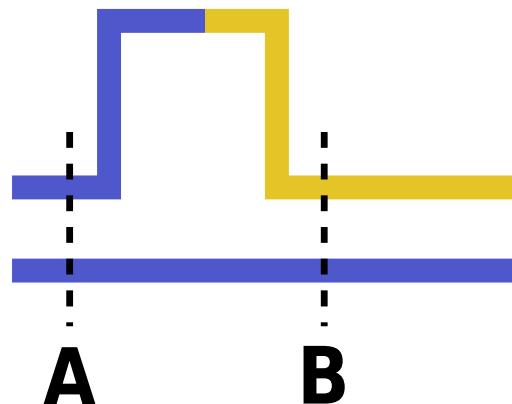


Phys. Rev. Lett. 100, 126802 (2008)

Time-dependent current



Oscillation frequency: eV_b/h



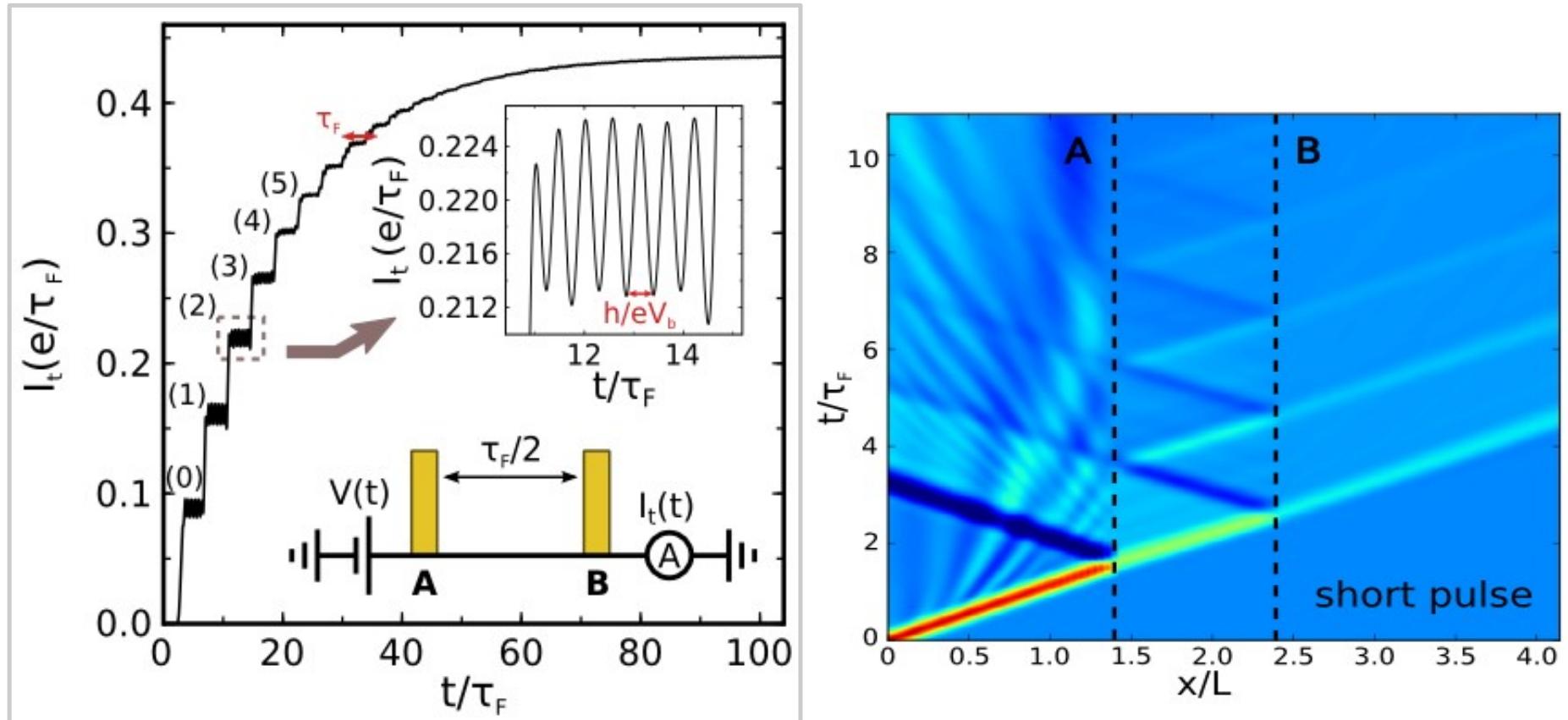
$$a_{\uparrow} e^{ikL_{\uparrow}}$$

$$a_{\downarrow} e^{ikL_{\downarrow}} \times e^{-ieV_b t}$$

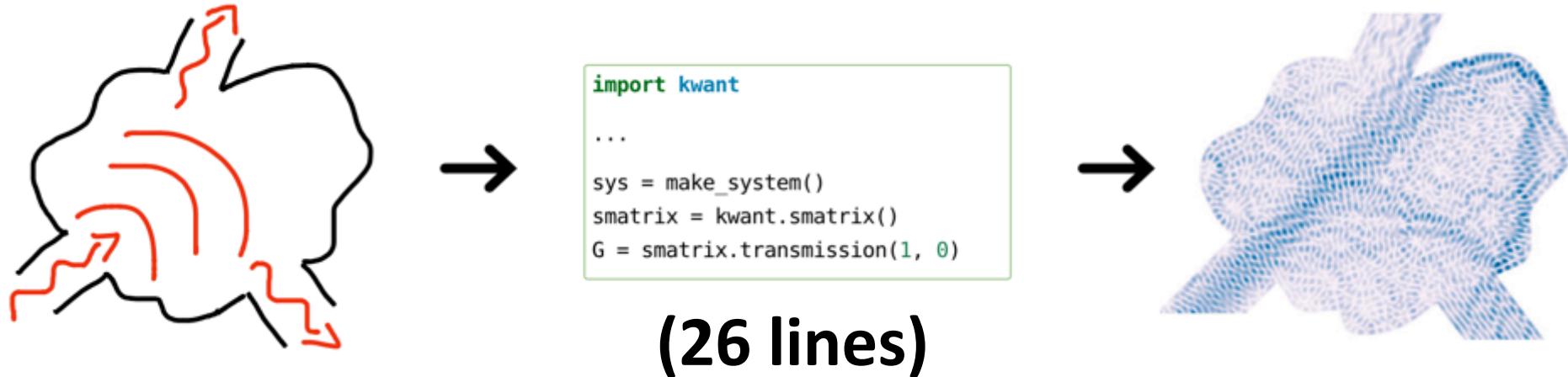
$$I_1(t) = \frac{e^2 V_b}{h} D_A D_B + \frac{e}{\pi \tau_F} \sqrt{D_A D_B R_A R_B} \cos \left(\frac{eV_b}{\hbar} t + \phi \right)$$

Dynamical control of an interference pattern

Fabry-Perot interferometer



Kwant is our new software for simulating quantum transport.



Kwant paradigms:

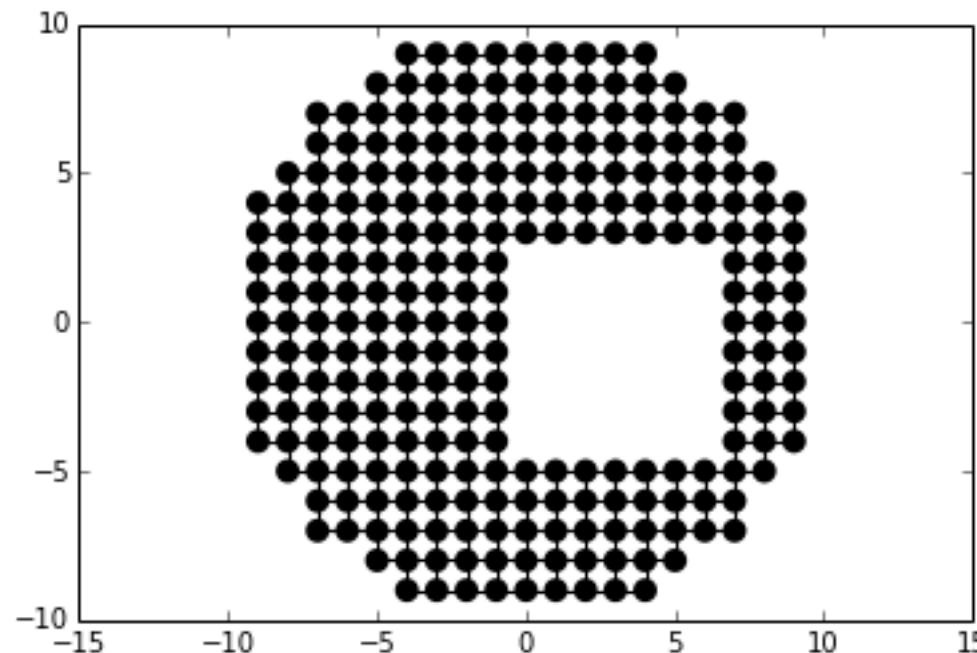
- Low entrance cost for physicists.
- Spend your time on physics not writing code (let the computer do the book keeping)
- Make the computer aware of physics concepts (symmetries, lattices,...)
- Simultaneously Fast AND Flexible (faster than recursive Green's function: nested dissection)

A collaboration with Christoph Groth, Anton Akhmerov (Delft) and Michael Wimmer (Delft)

An open source software available on Linux, Mac and even Windows.

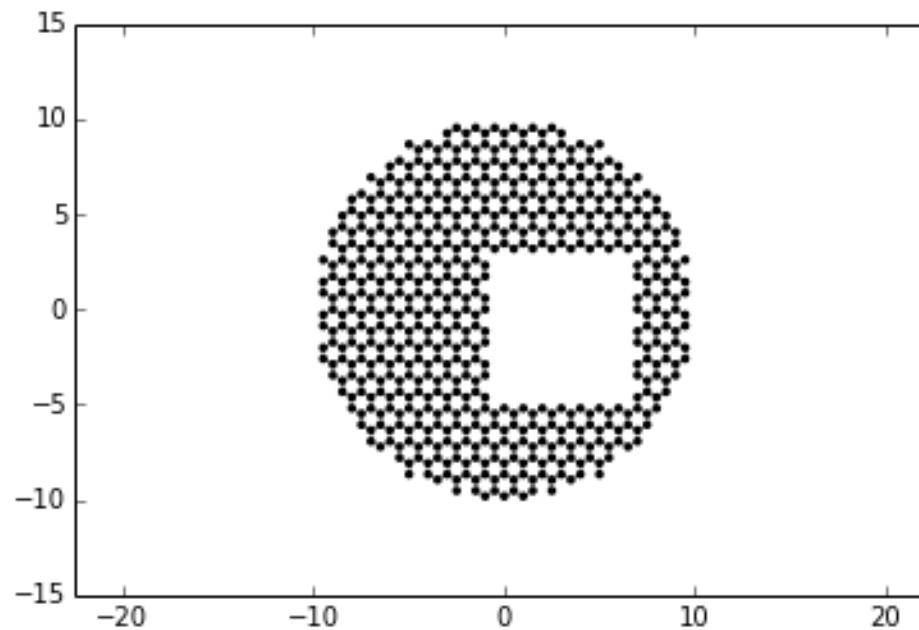
<http://kwant-project.org>

```
In [3]: import kwant
lat= kwant.lattice.square()
def circle(R): return R[0]** 2 + R[1]** 2 < 100
def square(R): return abs(R[0]-3) < 4 and abs(R[1]+1) < 4
H=kwant.Builder()
H[lat.shape(circle,(0,0))]=0
H[lat.neighbors()]=1
del H[lat.shape(square,(0,0))]
kwant.plot(H,site_lw=0.3);
```



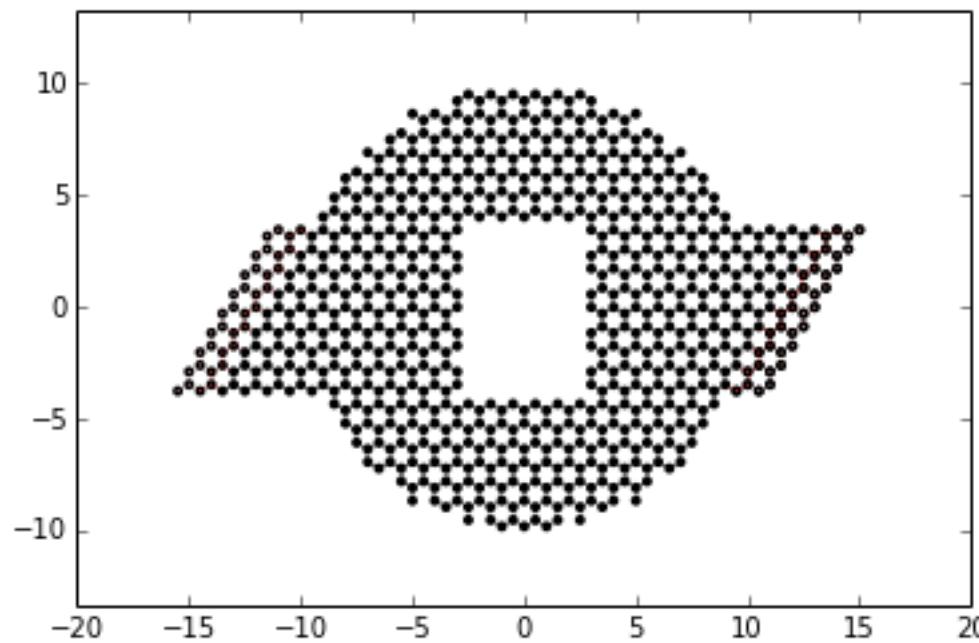
WRITE YOUR HAMILTONIAN AS YOU DO ON THE BLACKBOARD

```
In [2]: import kwant
lat= kwant.lattice.honeycomb()
def circle(R): return R[0]** 2 + R[1]** 2 < 100
def square(R): return abs(R[0]-3) < 4 and abs(R[1]+1) < 4
H=kwant.Builder()
H[lat.shape(circle,(0,0))]=0
H[lat.neighbors()]=1
del H[lat.shape(square,(0,0))]
kwant.plot(H,site_lw=0.3);
```



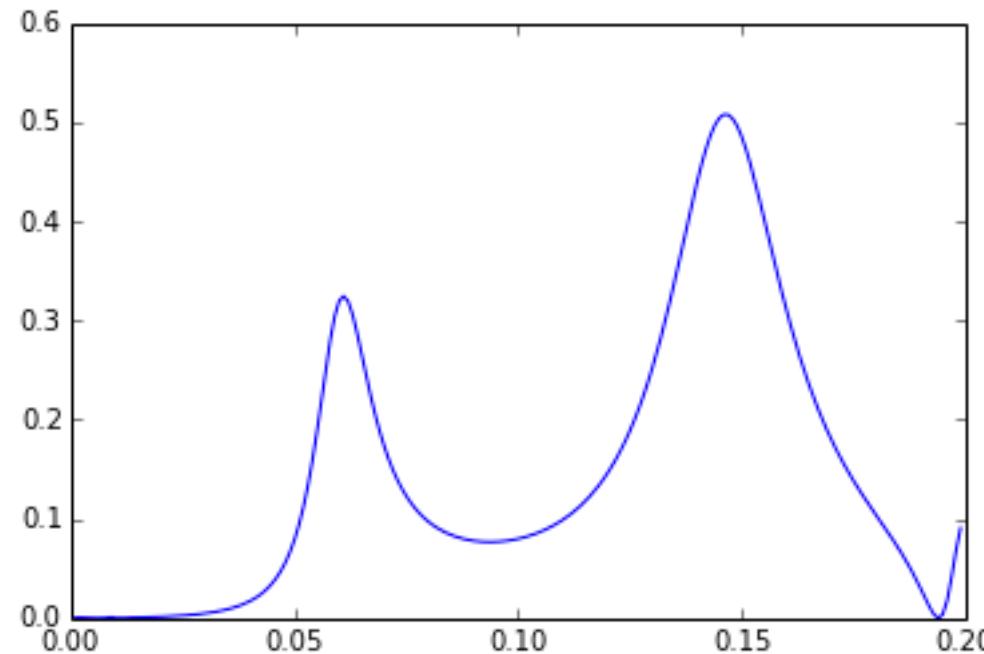
KWANT SUPPORT MANY PHYSICAL CONCEPTS SUCH AS BRAVAIS LATTICE

```
In [8]: sym=kwant.TranslationalSymmetry(lat.vec((1,0)))
def lead_shape(R): return abs(R[1]) < 4 and abs(R[0]) <3
Hlead =kwant.Builder(sym)
Hlead[lat.shape(lead_shape,(0,0) )]=0
Hlead[lat.neighbors()]=1
H.attach_lead(Hlead)
H.attach_lead(Hlead.reversed())
kwant.plot(H,site_lw=0.3);
```



AND TRANSLATIONAL SYMMETRY. THE SYSTEM IS NOW INFINITE

```
In [5]: from matplotlib import pyplot
Hf=H.finalized()
data = []
energies = [ix*0.001 for ix in xrange(1,200)]
for ee in energies:
    smatrix = kwant.smatrix(Hf, ee)
    data.append(smatrix.transmission(1, 0))
pyplot.plot(energies, data);
```

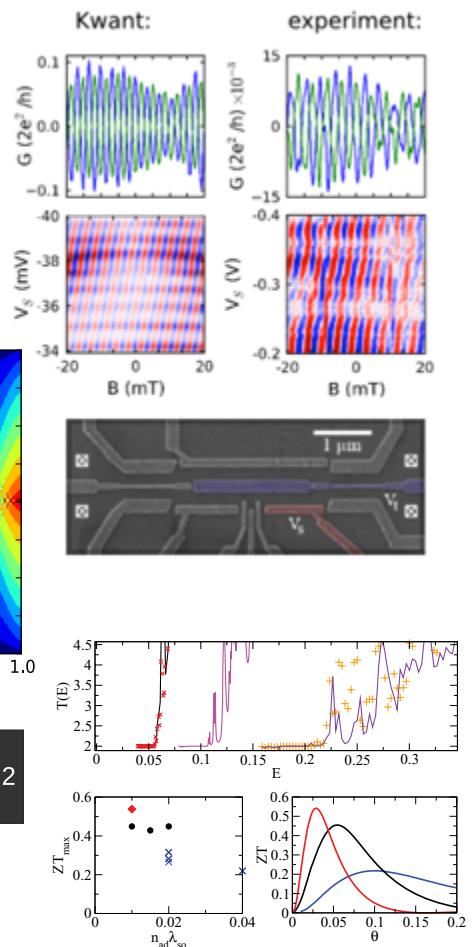
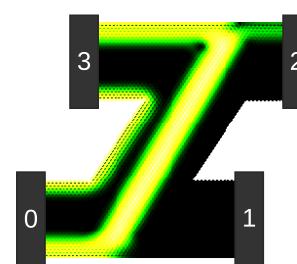
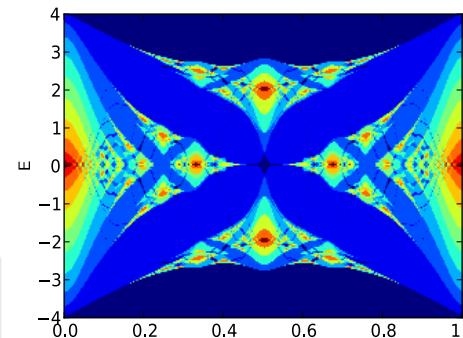
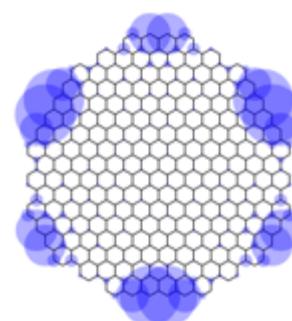
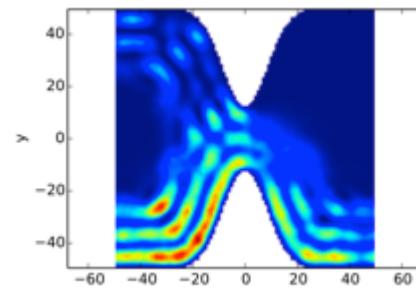
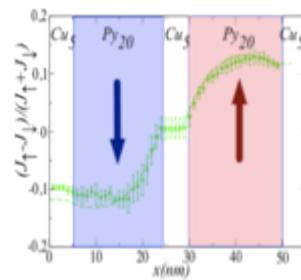
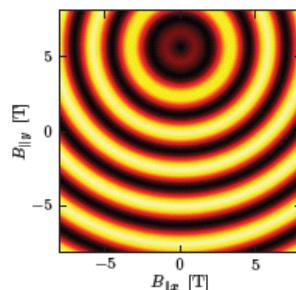
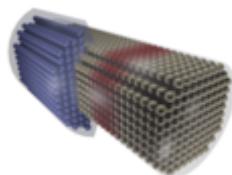


...AND WE CAN START CALCULATING OBSERVABLES SUCH AS THE CONDUCTANCE

- Arbitrary geometry
- Arbitrary internal degrees of freedom (spin, nambu, orbitals...)
- Function as values

```
def V(R): return 0.01*R.pos[0]*(R.pos[1]+0.5)
H[lat.shape(circle,(0,0))]=V
```

- Multi-terminal
- Arbitrary dimension

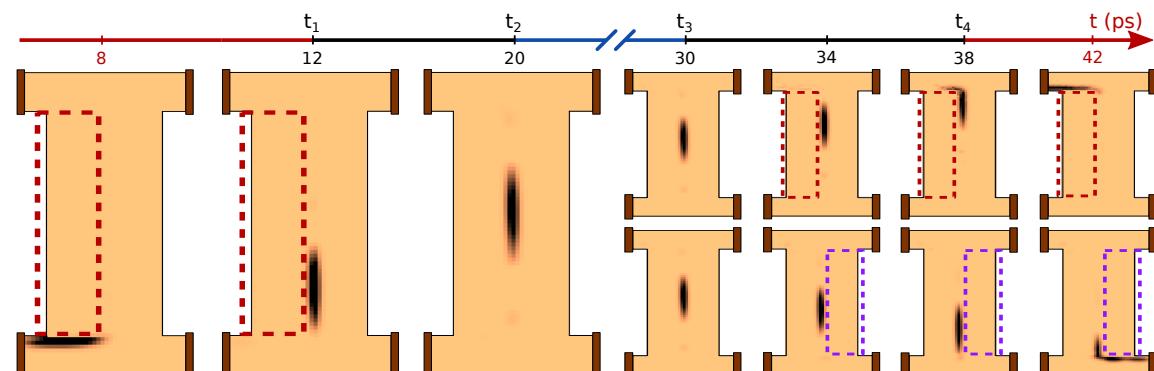
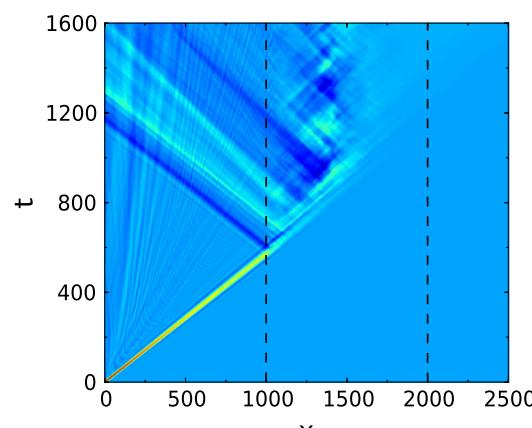
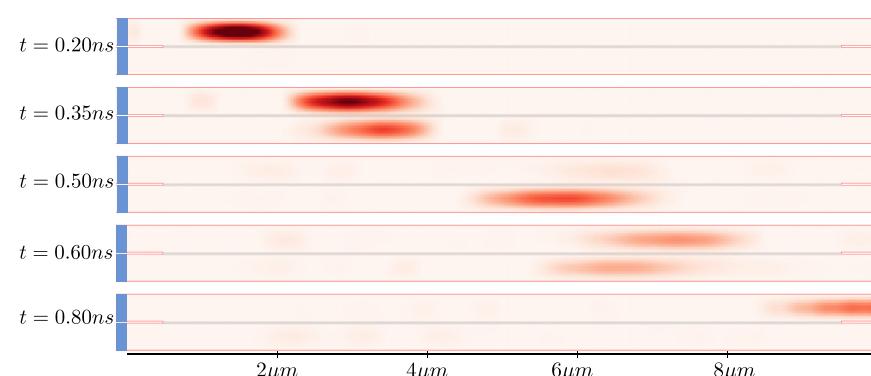
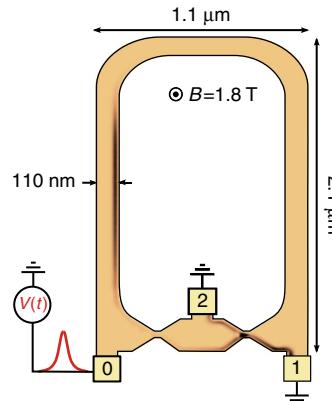


```
def V(R): return 0.01*R.pos[0]*(R.pos[1]+0.5)
H[lat.shape(circle,(0,0) )]=V
```

Becomes:

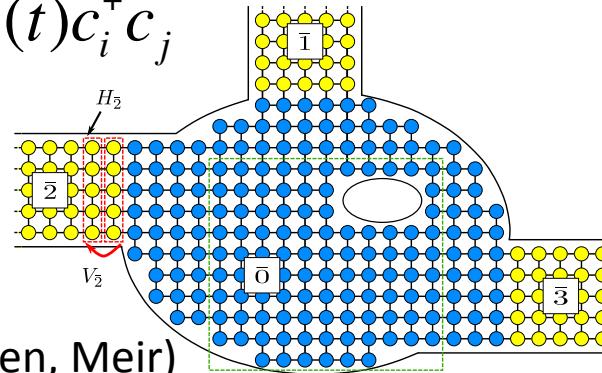
```
def V(t,R): return 0.01*R.pos[0]*(R.pos[1]+0.5)+0.2*cos(w*t)
H[lat.shape(circle,(0,0) )]=V
```

Will be open source but not ready yet.



Mathematical aspects: Non Equilibrium Green Function **versus** Scattering theory **versus** partitionless approach

$$\hat{H}(t) = \sum_{ij} H_{ij}(t) c_i^+ c_j$$



Non Equilibrium Green's Function (NEGF) approach

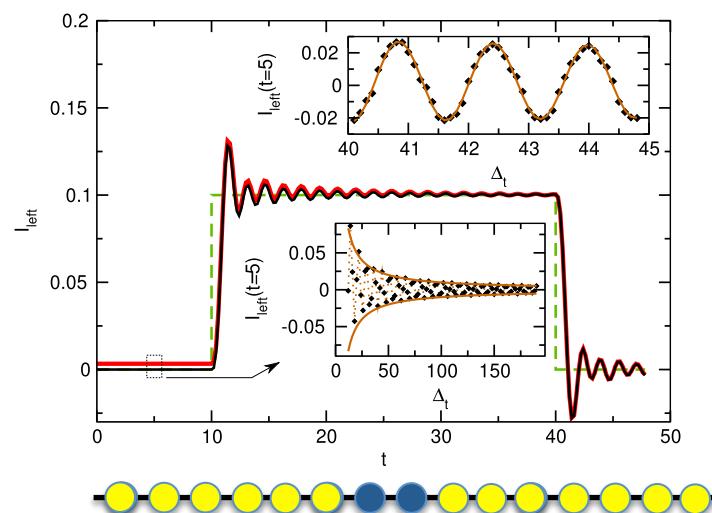
- Write Dyson equation in Keldysh space
- Integrate out the leads degrees of freedom
- One finds the standard framework (Jauho, Wingreen, Meir)

$$i\partial_t G(\vec{r}, t, \vec{r}', t') = H(t)G(\vec{r}, t, \vec{r}', t') + \int du d\vec{r}'' \Sigma(\vec{r}, t, \vec{r}'', u)G(\vec{r}'', u, \vec{r}', t')$$

$$G^<(\vec{r}, t, \vec{r}', t') = \int du dv d\vec{r}'' d\vec{r}''' G(\vec{r}, t, \vec{r}'', u) \Sigma^<(\vec{r}''', u, \vec{r}''', v) G^+(\vec{r}''', v, \vec{r}', t')$$

Computationally prohibitive:

- CPU=t²L⁷(3D)
- One value of t per calculation
- Huge memory footprint
- Large time just to recover the initial stationary solution
(before switching on any time-dependent field)
- Algorithms easily unstable.



Mathematical aspects: Partitionless approach

System at equilibrium

$$H_0 \Psi_{\alpha E}^{st} = E \Psi_{\alpha E}^{st}$$

$$\vec{I}_{EE}(\vec{r}, t) = \text{Im} \left[\Psi_{\alpha E}^*(\vec{r}, t) \vec{\nabla} \Psi_{\alpha E}(\vec{r}, t) \right]$$

$$\langle \hat{\vec{I}}(\vec{r}, t) \rangle = \sum_{\alpha} \int \frac{dE}{2\pi} f_{\alpha}(E) \vec{I}_{EE}(\vec{r}, t)$$

Pauli

$$\langle \hat{\vec{I}}(\vec{r}, t) \cdot \hat{\vec{I}}(\vec{r}', t') \rangle - \langle \hat{\vec{I}}(\vec{r}, t) \rangle \cdot \langle \hat{\vec{I}}(\vec{r}', t') \rangle = \sum_{\alpha, \beta} \int \frac{dE}{2\pi} \int \frac{dE'}{2\pi} f_{\alpha}(E) (1 - f_{\beta}(E')) \vec{I}_{EE'}(\vec{r}, t) \cdot \vec{I}_{EE'}^*(\vec{r}', t')$$

Wave function fully equivalent to NEGF

$$G^R(t, t') = -i\theta(t - t') \int \frac{dE}{2\pi} \sum_{\alpha} \Psi_{\alpha E}(t) \Psi_{\alpha E}^\dagger(t')$$

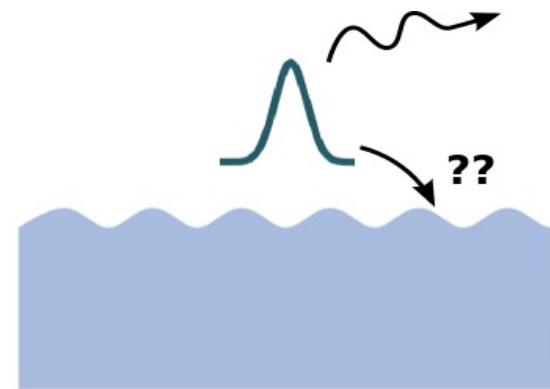
$$G^<(t, t') = \sum_{\alpha} \int \frac{dE}{2\pi} i f_{\alpha}(E) \Psi_{\alpha E}(t) \Psi_{\alpha E}^\dagger(t')$$

Perturbation switched on

$$i\partial_t \Psi_{\alpha E}(t) = H(t) \Psi_{\alpha E}(t)$$

time

t=0



Mathematical aspects: Non Equilibrium Green Function **versus** Scattering theory **versus** partitionless approach

$$\hat{H}(t) = \sum_{ij} H_{ij}(t) c_i^+ c_j$$

$$H(t) = H_0 + \Delta(t)$$

Going to a practical scheme:

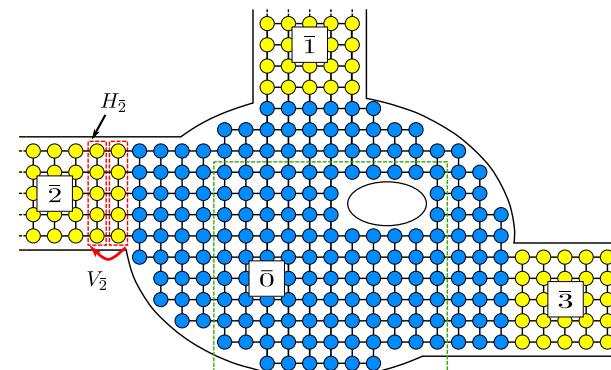
$$\Psi_E(\vec{r}, t) = \bar{\Psi}_E(\vec{r}, t) + \Psi_E^K(\vec{r}) e^{-iEt}$$

$$i\partial_t \bar{\Psi}_E(\vec{r}, t) = \overset{\text{SCHRODINGER}}{H(t)} \bar{\Psi}_E(\vec{r}, t) + \overset{\text{SOURCE}}{\Delta(t)} \Psi_E^K(\vec{r}) e^{-iEt} + \overset{\text{SINK}}{i\Sigma(\vec{r})} \bar{\Psi}_E(\vec{r}, t)$$

$$\vec{I}(\vec{r}, t) = \int dE \operatorname{Im} \Psi_E(\vec{r}, t) \vec{\nabla} \Psi_E(\vec{r}, t) f(E)$$

Almost computationally easy:

- CPU=t L³(3D)
- All values of t at once
- Small memory footprint L³
- Start from the exact stationary solution
- Stable differential equation
- Easily amenable to analytical solutions
- Currently we solve systems with more than 10⁵ sites
- Algorithm very parallel.



Conclusion

Quantum electronics in the time domain is only starting to emerge

Time resolved simulations are now relatively easy

Even the simplest cases provide nice new concepts