```
% Ten-spin simulation.
%
% i.kuprov@soton.ac.uk

function hello_again_world()

% Pauli matrices
sigma_x=sparse([0 1/2; 1/2 0]);
sigma_y=sparse([0 -1i/2; 1i/2 0]);
sigma_z=sparse([1/2 0; 0 -1/2]);
unit=sparse([1 0; 0 1]);

% Number of spins
nspins=7;

% Cell arrays of operators
Lx=cell(1,nspins); Ly=cell(1,nspins); Lz=cell(1,nspins);
for n=1:nspins
    Lx_current=1; Ly_current=1; Lz_current=1;
    for k=1:nspins
        if k==n
            Lx_current=kron(Lx_current,sigma_x);
            Ly_current=kron(Ly_current,sigma_y);
            Lz_current=kron(Lz_current,sigma_z);
        else
            Lx_current=kron(Lx_current,unit);
            Ly_current=kron(Ly_current,unit);
            Lz_current=kron(Lz_current,unit);
        end
    end
    Lx{n}=Lx_current; Ly{n}=Ly_current; Lz{n}=Lz_current;
end

% Hamiltonian operator
zeeman_freqs=2*pi*1000*randn(1,nspins);
scalar_couplings=100*pi*randn(nspins,nspins);

% Preallocate Hamiltonian array
H=spalloc(2^nspins,2^nspins,(nspins^2)*(2^nspins));

% Zeeman interactions
for n=1:nspins
    H=H+zeeman_freqs(n)*Lz{n};
end

% Scalar couplings
for n=1:nspins
    for k=1:nspins
        if n~=k
            H=H+scalar_couplings(n,k)*(Lx{n}*Lx{k}+Ly{n}*Ly{k}+Lz{n}*Lz{k});
        end
    end
end

% Initial state
```

```
rho=spalloc(2^nspins,2^nspins,(nspins^2)*(2^nspins));
for n=1:nspins
    rho=rho+Lz{n};
end

% Detection state
coil=spalloc(2^nspins,2^nspins,(nspins^2)*(2^nspins));
for n=1:nspins
    coil=coil+Lx{n}+1i*Ly{n};
end

% Pulse Hamiltonian
Hp=spalloc(2^nspins,2^nspins,(nspins^2)*(2^nspins));
for n=1:nspins
    Hp=Hp+Ly{n};
end

% Build propagators
P_pulse=sparse(expm(-1i*Hp*(pi/2)));
time_step=1/normest(H);
P_evol=sparse(expm(-1i*H*time_step));

% Clean up propagators
P_pulse=P_pulse.*(abs(P_pulse)>1e-6);
P_evol=P_evol.*(abs(P_evol)>1e-6);

% Simulation, stage 1: pulse
rho=P_pulse*rho*P_pulse';

% Simulation, stage 2: evolution
nsteps=2048;       % number of steps in the simulation
fid=zeros(2048,1); % preallocate the array
for n=1:nsteps
    fid(n)=trace(coil'*rho);
    rho=P_evol*rho*P_evol';
    disp(n);
end

% Apodization
window_function=exp(-5*linspace(0,1,2048))';
fid=fid.*window_function;

% Fourier transform with zerofill
spectrum=fftshift(fft(fid,8196));

% Plotting
plot(real(spectrum));

end
```