Exercise 1

$$\begin{cases} \partial_t V(S,t) - \frac{\sigma^2}{2} S^2 \partial_{SS} V(S,t) - r S \partial_S V(S,t) + r V(S,t) = 0 & \text{in } [0, S_{max}) \times (0,T] \\ V(S_{max}, t) = 0 & \\ V(S,0) = G(S) = \max\{K-S, 0\} & \end{cases} \quad (1)$$

a) Take $\nu = \partial_\sigma V$ and differentiate the previous equation.

- $\partial_\sigma \left(\partial_t V(S,t) - \frac{\sigma^2}{2} S^2 \partial_{SS} V(S,t) - r S \partial_S V(S,t) + r V(S,t) \right) = 0$

$\Rightarrow \partial_t \partial_\sigma V(S,t) - \sigma S^2 \partial_{SS} V(S,t) - \frac{\sigma^2}{2} S^2 \partial_{SS} \partial_\sigma V(S,t) - r S \partial_S \partial_\sigma V(S,t) +$

$+ r \partial_\sigma V(S,t) = 0$

$\Rightarrow \partial_t \nu(S,t) - \frac{\sigma^2}{2} S^2 \partial_{SS} \nu(S,t) - r S \partial_S \nu(S,t) + r \nu(S,t) = \sigma S^2 \partial_{SS} V(S,t)$

- $\partial_\sigma V(S_{max}, t) = 0 \Rightarrow \nu(S_{max}, t) = 0$

- $\partial_\sigma V(S,0) = \partial_\sigma G(S) = 0 \Rightarrow \nu(S,0) = 0$

Then, we obtain the PDE

$$\begin{cases} \partial_t \nu(S,t) - \frac{\sigma^2}{2} S^2 \partial_{SS} \nu(S,t) - r S \partial_S \nu(S,t) + r \nu(S,t) = \sigma S^2 \partial_{SS} V(S,t) & \\ \nu(S_{max}, t) = 0 & \text{in } [0, S_{max}) \times (0,T] \\ \nu(S,0) = 0 & \end{cases}$$

This is a parabolic PDE since the coefficients of the differential operator relating to the $\partial_{tt} \nu$ and $\partial_S \partial_t \nu$ are equal to 0. Moreover, unlike equation (1), the PDE for Vega is not homogenous.

Take Home Exam 4: exercise 1 b) c) g) h)

Alessandro La Farciola

December 1, 2022

In this file there are the Matlab codes for *Exercises b) c) d).*

Exercise b)

(i) Compute the solution of the first PDE, namely V(S,t).

(ii) Here, I want to approximate with finite differences the RHS of the PDE for ν. In particular, I use

$$D_h^2 u(x) = \frac{u(x+h) - 2u(x) + u(x-h)}{h^2}.$$

Since I have the vector V that is the numerical solution of the first PDE, for each $i = 2, \ldots, end{-}1$

$$D_h^2 V(i) = \frac{V(i+1) - 2V(i) + V(i-1)}{h^2}, \quad where \;\; h = \frac{S_{max}}{N_x}.$$

(iii) Plot the exact solution computed with *blsvega* and the one computed before.

The maximum error of the approximation is 0.0015 and the following figure shows the result obtained.

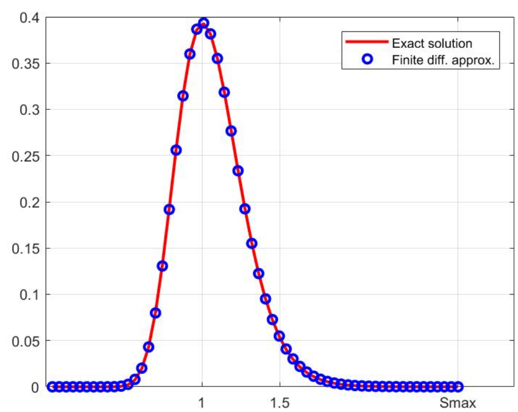

Figure 1: Comparison the exact solution and the approximation of *vega* .

Here it is the code for the solution of the 1.b.

```
%(i)
sigma=0.2; r=0.015; K=1; T=1; tol=1e-6; Nx=60; Nt=100;
initial_cond=@(S) max(K-S,0);
%forcing
rhs= @(x,t) 0*x;
% compute the right boundary
Smax = K*exp(sigma^2*T/2-sigma*sqrt(T)*norminv(tol/K));
bc_right=@(t) 0*t;
theta=0.5;

% numerical solution for V
[V,FD_grid,time_steps] = bs_timestepping_const_sigma(sigma,r,rhs,bc_right
    ,initial_cond,Smax,Nx,T,Nt,theta);

%discretization of rhs
h = Smax/Nx;
ddS=@(t) (V(3:end,t+1)-2*V(2:end-1,t+1)+V(1:end-2,t+1))/h^2;
rhs_nu=@(S,t) [S(1) sigma*S(2:end).^2.*ddS(t)'];
initial_cond_nu=@(S) 0*S;
% numerical solution for nu
[nu,FD_grid_nu,time_steps] = bs_timestepping_const_sigma_vega(sigma,r,
    rhs_nu,bc_right,initial_cond_nu,Smax,Nx,T,Nt,theta);

% exact solution with blsvega
Vexact=[];
for v=FD_grid(2:end)
      Put = blsvega(v, K, r, T, sigma);
      Vexact=[Vexact Put];
end
%comparison with blsvega
figure
set(gca,'FontSize',20)
plot(FD_grid(2:end),Vexact,'r','LineWidth',2,'DisplayName','Exact
    solution')
hold on
plot(FD_grid(2:end),nu(2:end,end),'ob','LineWidth',2,'DisplayName','
    Finite diff. approx.')
grid on
legend show
set(legend,'Location','NorthEast')
set(gca,'XTick',[1 1.5 Smax])
set(gca,'XTickLabel',{'1','1.5','Smax'})
% difference between exact solution and finite differences approximation
max(abs(nu(2:end,end)'-Vexact))
```

In the previous code, I used two functions:

- *bs_timestepping_const_sigma*, that is the one written in Exercise 1 of Homework 8.
- *bs_timestepping_const_sigma_vega*, that is the same of the previous one, but at line 108 I changed *f_new = forcing(inner_grid,t_new)'* with *f_new = forcing(inner_grid,tn)'* (This func-

tion is included in the matlab file *TH4_b.m*).

Exercise c)

In this section, I firstly present the script to obtain the plot to show the rate of convergence.

```
%1.c

sigma=0.2; r=0.015; K=1; T=1; tol=1e-6;
initial_cond=@(S) max(K-S,0);
%forcing
rhs= @(x,t) 0*x;
% compute the right boundary
Smax = K*exp(sigma^2*T/2-sigma*sqrt(T)*norminv(tol/K));
bc_right=@(t) 0*t;
theta=0.5;


%check of convergence
err=zeros(6,1);
h=zeros(6,1);
for i=0:5
    Nx=10*2^i;
    Nt=0.6*Nx;
    % finite differences solver
    [V,FD_grid,time_steps] = bs_timestepping_const_sigma(sigma,r,rhs,bc_right,initial_cond,Smax,Nx,T,Nt,theta);
    h(i+1)=Smax/Nx;
    ddS=@(t) (V(3:end,t+1)-2*V(2:end-1,t+1)+V(1:end-2,t+1))/h(i+1)^2;
    rhs_nu=@(S,t) [S(1) sigma*S(2:end).^2.*ddS(t)'];
    initial_cond_nu=@(S) 0*S;
    [nu,FD_grid_nu,time_steps] = bs_timestepping_const_sigma_vega(sigma,r,rhs_nu,bc_right,initial_cond_nu,Smax,Nx,T,Nt,theta);
    % exact solution with blsprice
    Vexact=[];
    for v=FD_grid(2:end)
        Put = blsvega(v, K, r, T, sigma);
        Vexact=[Vexact Put];
    end
    err(i+1)=max(abs(nu(2:end,end)'-Vexact));
end
figure
set(gca,'FontSize',20)
loglog(h,h*err(1)/h(1),'-.','LineWidth',2,'MarkerSize',10,'DisplayName','linear')
hold on
loglog(h,h.^2*err(1)/h(1)^2,'--','LineWidth',2,'MarkerSize',10,'DisplayName','quadratic')
```

```matlab
hold on
loglog(h,err,'-xk','LineWidth',1.5,'MarkerSize',8,'DisplayName','errors')
grid on
legend show
set(legend,'Location','SouthEast')
```

Now, the following figure shows the plot obtained with the previous script

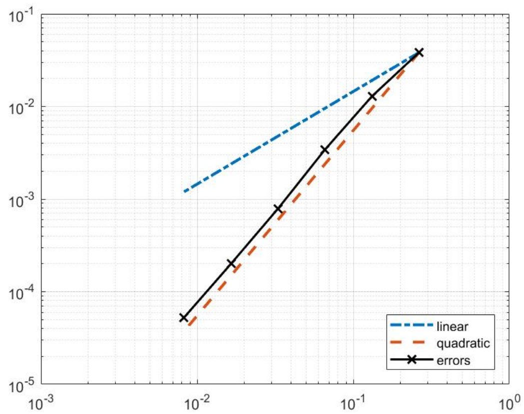

Figure 2: Order of convergence in a log-log plot.

As we can see from the previous plot, the order of convergence is quadratic.

Exercise d)

(i) In this section, I firstly present the script to *directly approximate* ν using the central difference scheme for the derivative with respect to σ.

```matlab
sigma=0.2; r=0.015; K=1; T=1; tol=1e-6;
initial_cond=@(S) max(K-S,0);
%forcing
rhs= @(x,t) 0*x;
% compute the right boundary
Smax = K*exp(sigma^2*T/2-sigma*sqrt(T)*norminv(tol/K));
bc_right=@(t) 0*t;
theta=0.5;
Nx=60;
Nt=100;

%1.d

%(i)
%central finite difference scheme to directly approximate nu
deltasigma= [0.0005; 0.001; 0.01; 0.02; 0.03; 0.05];
```

```
figure
set(gca,'FontSize',13)
for i=6:-1:1
    [V,FD_grid,time_steps] = bs_timestepping_const_sigma(sigma+deltasigma(i),r,rhs,bc_right,initial_cond,Smax,Nx,T,Nt,theta);
    [V1,FD_grid,time_steps] = bs_timestepping_const_sigma(sigma-deltasigma(i),r,rhs,bc_right,initial_cond,Smax,Nx,T,Nt,theta);
    nu1=(V(:,end)-V1(:,end))/(2*deltasigma(i));
    hold on
    plot(FD_grid(1:end),nu1(1:end,end),'DisplayName',sprintf('Approx ds=%.4f',deltasigma(i)))
end
Vexact=[];
for v=FD_grid(2:end)
      Put = blsvega(v, K, r, T, sigma);
      Vexact=[Vexact Put];
end
hold on
plot(FD_grid(2:end),Vexact,'r','LineWidth',0.5,'DisplayName','Exact solution')
grid on
legend show
set(legend,'Location','NorthEast')
set(gca,'XTick',[1 1.5 Smax])
set(gca,'XTickLabel',{'1','1.5','Smax'})


%(ii)

%check of convergence
err=zeros(6,1);
h=zeros(6,1);
for i=0:5
    Nx=10*2^i;
    Nt=0.6*Nx;
    % finite differences solver
    [V,FD_grid,time_steps] = bs_timestepping_const_sigma(sigma,r,rhs,bc_right,initial_cond,Smax,Nx,T,Nt,theta);
    h(i+1)=Smax/Nx;
    ddS=@(t) (V(3:end,t+1)-2*V(2:end-1,t+1)+V(1:end-2,t+1))/h(i+1)^2;
    rhs_nu=@(S,t) [S(1) sigma*S(2:end).^2.*ddS(t)'];
    initial_cond_nu=@(S) 0*S;
    [nu,FD_grid_nu,time_steps] = bs_timestepping_const_sigma_vega(sigma,r,rhs_nu,bc_right,initial_cond_nu,Smax,Nx,T,Nt,theta);
    % exact solution with blsprice
    Vexact=[];
    for v=FD_grid(2:end)
          Put = blsvega(v, K, r, T, sigma);
          Vexact=[Vexact Put];
```

```
    end
    err(i+1)=max(abs(nu(2:end,end)'-Vexact));
end
figure
set(gca,'FontSize',20)
loglog(h,h*err(1)/h(1),'-.','LineWidth',2,'MarkerSize',10,'DisplayName','linear')
hold on
loglog(h,h.^2*err(1)/h(1)^2,'--','LineWidth',2,'MarkerSize',10,'DisplayName','quadratic')
hold on
loglog(h,err,'-xk','LineWidth',1.5,'MarkerSize',8,'DisplayName','errors')
grid on
legend show
set(legend,'Location','SouthEast')

%adding the new errors for the central finite differe scheme for all deltasigmas
for j=6:-1:1
    for i=0:5
    Nx=10*2^i;
    Nt=0.6*Nx;
    h(i+1)=Smax/Nx;
    % finite differences solver
    [V,FD_grid,time_steps] = bs_timestepping_const_sigma(sigma+deltasigma(j),r,rhs,bc_right,initial_cond,Smax,Nx,T,Nt,theta);
    [V1,FD_grid,time_steps] = bs_timestepping_const_sigma(sigma-deltasigma(j),r,rhs,bc_right,initial_cond,Smax,Nx,T,Nt,theta);
    nu1=(V(:,end)-V1(:,end))/(2*deltasigma(j));
    % exact solution with blsprice
    Vexact=[];
    for v=FD_grid(2:end)
        Put = blsvega(v, K, r, T, sigma);
        Vexact=[Vexact Put];
    end
    err(i+1)=max(abs(nu1(2:end,end)'-Vexact));
    end
    hold on
    loglog(h,err,'-x','LineWidth',1,'MarkerSize',6,'DisplayName',sprintf('err ds=%.4f',deltasigma(j)))
end
grid on
legend show
set(legend,'Location','SouthEast')
```

The following figure shows the results obtained.

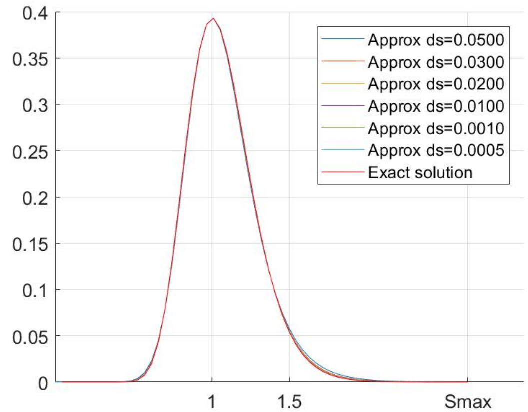

Figure 3: Approximation of *vega* with central difference scheme.

(ii) Now, I compare the methods just applied and the one used in part b) to approximate ν. In particular, I add the errors of the previous methods for each $\delta\sigma$ on the log-log plot obtained in part c).

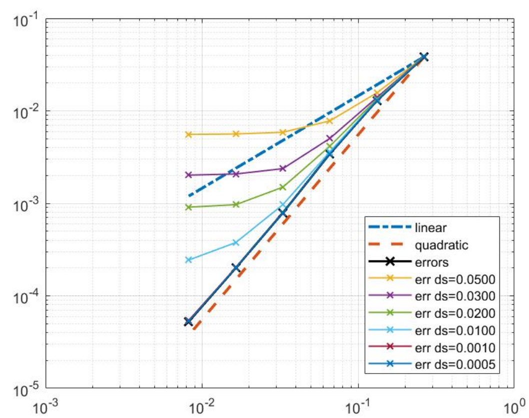

Figure 4: Approximation of *vega* with central difference scheme.

As we can see from the last plot, the errors for the methods implemented with the central difference scheme are higher than the ones computed in part b). Moreover, the rate of convergence of the error increases when $\delta\sigma$ decreases. In particular, for $\delta\sigma = 0.001$ and $\delta\sigma = 0.0005$ we reach the quadratic order of convergence as in part b).