Exam: Solutions

1. a) We select an arbitrary smooth function $u \in C_0^\infty((-1,1))$. Using integration by parts we obtain

$$\begin{aligned}\int_{-1}^{1} u'\varphi \, \mathrm{d}x &= \int_{-1}^{0} u'\varphi \, \mathrm{d}x + \int_{0}^{1} u'\varphi \, \mathrm{d}x \\ &= (u\varphi)\Big|_{-1}^{0} - \int_{-1}^{0} u\varphi' \mathrm{d}x + (u\varphi)\Big|_{0}^{1} - \int_{0}^{1} u\varphi' \mathrm{d}x \\ &= u(0)(\varphi_- - \varphi_+) - \int_{-1}^{1} u\varphi' \mathrm{d}x.\end{aligned}$$

We conclude that the first weak derivative of φ is given by $\varphi' + \delta_0(\varphi_+ - \varphi_+)$.

i) Above calculations imply that the first weak derivative of φ is equal to $\varphi' \in L^2((-1,1))$. Hence $\varphi \in H^1((-1,1))$.

ii) Above calculations and the fact that the Dirac functional δ_0 is not in $L^2((-1,1))$ prove that $\varphi \notin H^1((-1,1))$.

b) In the CIR model, r_t will always be non-negative. In the Vasicek model, r_t can also be negative.

c) i) The condition $\beta_+ > 1$ is required to control the truncation error in the case of call options, due to linear growth of the payoff. For put options the value $\beta_+ = 1$ the condition that

$$\int_{|z|>1} e^z k(z) \mathrm{d}z < \infty$$

is also satisfied using that $\alpha > 0$.

ii) The case $k(z) = 0$, $z \in \mathbb{R}$, means that the Lévy measure is zero. This implies that the stiffness matrix is sparse, because the infinitesimal generator of X is just a differential operator of second order. Numerically, linear systems are fast to solve. In the case $k(z) > 0$, $z \in \mathbb{R}$, the infinitesimal generator of X has an integration part, which implies that the stiffness matrix has in general no sparsity. Numerically, this means that linear systems are expensive to solve, which limits the space discretization, where speed and needed memory to store the matrix becomes challenging.

iii) For $\alpha = 2$, we cannot conclude under the made assumption that $\int_{\mathbb{R}} \min\{1, z^2\} k(z) \mathrm{d}z < \infty$. The assumptions $k(z) \le C/|z|^{1+\alpha}$ only implies that

$$\int_{|z|<1} z^2 k(z) \mathrm{d}z \le C \int_{|z|<1} \frac{1}{|z|} \mathrm{d}z = 2C \int_0^1 \frac{1}{z} \mathrm{d}z = -2C \lim_{z \to 0} \log(z) = \infty.$$

Please turn over.

d) There exist constants $C(T, \sigma), \gamma_1, \gamma_2 > 0$ that are independend of x such that for every $x \in (-R, R)$

$$\sup_{t \in (0,T)} |v_R(t,x) - v(t,x)| \leq Ce^{-\gamma_1 R + \gamma_2 |x|}.$$

e) The log-moneyness transformation yields that returned spot and option prices can be multiplied by K to achieve the corresponding result for the desired strike $K \neq 1$.

f) We expect h^2 and $N^{-2/3}$, due to the curse of dimension. The relationship is $N \simeq h^{-d}$, where d is the dimension of the spacial domain.

Continued overleaf.

2. **a)** The price of an European option with butterfly payoff is defined by

$$V(t,s) := \mathbb{E}[e^{-r(T-t)} g^{\mathrm{BF}}(S_T) | S_t = s], \quad t \in [0,T], s \in (0,\infty).$$

The price $V(t,s)$ satisfies the PDE

$$\begin{aligned} \partial_t V(t,s) + \frac{\sigma^2}{2} s^2 \partial_{ss} V(t,s) + rs\partial_s V(t,s) - rV(t,s) &= 0, && (t,s) \in [0,T) \times (0,\infty) \\ V(T,s) &= g^{\mathrm{BF}}(s), && s \in (0,\infty). \end{aligned}$$

b) Define $v(t,x) := V(T-t, e^x)$ for every $t \in [0,T]$ and every $x \in \mathbb{R}$. We obtain for every $(t,x) \in (0,T) \times \mathbb{R}$ that

$$\begin{aligned} \partial_t v(t,x) &= -\partial_t V(T-t, e^x), \\ \partial_x v(t,x) &= \partial_s V(T-t, e^x) e^x, \\ \partial_{xx} v(t,x) &= \partial_x (\partial_s V(T-t, e^x) e^x) = \partial_s V(T-t, e^x) e^x + \partial_{ss} V(T-t, e^x) e^{2x}. \end{aligned}$$

This implies that

$$s^2 \partial_{ss} V = \partial_{xx} v - \partial_x v.$$

Inserting into the result from the solution of **a)** implies that

$$\begin{aligned} \partial_t v - \frac{1}{2}\sigma^2 \partial_{xx} v + \left(\frac{1}{2}\sigma^2 - r\right) \partial_x v + rv &= 0 && \text{in } (0,T] \times (0,\infty), \\ v(0,x) &= g^{\mathrm{BF}}(e^x) && \text{for every } x \in \mathbb{R}. \end{aligned}$$

We define v_R to be solution of the following PDE on the truncated domain $(-R,R)$, where we assume that as minimal condition on R that $|\log(K)| < R$:

$$\begin{aligned} \partial_t v_R - \frac{1}{2}\sigma^2 \partial_{xx} v_R + \left(\frac{1}{2}\sigma^2 - r\right) \partial_x v_R + rv_R &= 0 && \text{in } (0,T] \times (-R,R), \\ v_R(0,x) &= g^{\mathrm{BF}}(e^x) && \forall x \in (-R,R), \\ v_R(t, \pm R) &= 0 && \forall t \in (0,T]. \end{aligned}$$

c) We have the following representation for u and v: $u(t,x) = \sum_{i=1}^N u_i(t) b_i(x)$ and $v(x) = \sum_{i=1}^N v_i b_i(x)$. Then, the discrete variational formulation is given for $\underline{\mathbf{u}}(t) =$

$(u_1(t), \ldots, u_N(t))^T$ and every $\underline{\mathbf{v}} = (v_1, \ldots, v_N)^T \in \mathbb{R}^N$ by

$$\sum_{i,j}(\partial_t u_i(t) b_i, v_j\, b_j)_{L^2(G)} + \frac{1}{2}\sigma^2 \sum_{i,j}(u_i(t) b_i', v_j\, b_j')_{L^2(G)}$$
$$+ (\frac{\sigma^2}{2} - r) \sum_{i,j}(u_i(t) b_i', v_j\, b_j)_{L^2(G)}$$
$$+ r \sum_{i,j}(u_i(t) b_i, v_j\, b_j)_{L^2(G)} = 0$$
$$\sum_{i,j} \partial_t u_i(t)(b_i\, b_j)_{L^2(G)} v_j + \frac{1}{2}\sigma^2 \sum_{i,j} u_i(t)(b_i'\, b_j')_{L^2(G)} v_j$$
$$+ (\frac{\sigma^2}{2} - r) \sum_{i,j} u_i(t)(b_i'\, b_j)_{L^2(G)} v_j$$
$$+ r \sum_{i,j} u_i(t)(b_i\, b_j)_{L^2(G)} v_j = 0$$

Therefore, setting $\mathbf{M}_{i,j} = (b_j, b_i)_{L^2(G)}$ and $\mathbf{S}_{i,j} = (b_j', b_i')_{L^2(G)}$ and $\mathbf{B}_{i,j} = (b_j', b_i)_{L^2(G)}$ and $\mathbf{A} = \frac{1}{2}\sigma^2 \mathbf{S} + (\frac{\sigma^2}{2} - r)\mathbf{B} + r\mathbf{M}$, we get

$$\mathbf{M}\partial_t \underline{\mathbf{u}}(t) + \mathbf{A}\underline{\mathbf{u}}(t) = 0.$$

We discretize in time with the θ-scheme and get ($\theta \in [0, 1]$)

$$\mathbf{M}\frac{\underline{\mathbf{u}}^m - \underline{\mathbf{u}}^{m-1}}{k} + (\theta \mathbf{A}\underline{\mathbf{u}}^m + (1-\theta)\mathbf{A}\underline{\mathbf{u}}^{m-1}) = 0$$

which leads to

$$(\mathbf{M} + k\theta\mathbf{A})\underline{\mathbf{u}}^m = (\mathbf{M} - k(1-\theta)\mathbf{A})\underline{\mathbf{u}}^{m-1}.$$

Here, $\underline{\mathbf{u}}^m = \underline{\mathbf{u}}(t_m)$. So we have for the variational form in matrix formulation: Find $\underline{\mathbf{u}}^m \in \mathbb{R}^N$ such that for all $m = 1, \ldots, M$

$$(\mathbf{M} + k\theta\mathbf{A})\underline{\mathbf{u}}^m = (\mathbf{M} - k(1-\theta)\mathbf{A})\underline{\mathbf{u}}^{m-1}$$

and $\underline{\mathbf{u}}^0 = \underline{\mathbf{u}}(0)$.

d) See the following code, which already includes the code for the American option with butterfly payoff below:

```
% MAIN_BSAMPUT computes value of an American put option
%    using the Black-Scholes model

clear all;
close all;

%-------------------------------------------------------------------------
%  Set Parameters
%-------------------------------------------------------------------------

% number of nodes
n = 2.^(5:9)'-1;
```

Continued overleaf.

```
% domain (-R,R)
R = 5;
% maturity
T = 1;
% strikes
K0 = 0.5;
K2 = 1.5;
K1 = (K0+K2)/2;

% interest rate
r =  0.005;
% volatiliy
sigma = 0.2;

theta = 0.5;
payoff_HANDLE = @(x) max(0, exp(x)-K0) - 2*max(0, exp(x)-K1) + max(0, exp(x)-K2);

SOL_HANDLE   =@(t,x) bs_formula_C(exp(x),t,K0,r,sigma) ...
    - 2 * bs_formula_C(exp(x),t,K1,r,sigma)...
    + bs_formula_C(exp(x),t,K2,r,sigma) ;


% loop over mesh points
errorL2L2 = zeros(length(n),1);

for i = 1:length(n)

    %----------------------------------------------------------------------
    %  Discretization
    %----------------------------------------------------------------------

    % mesh size
    h = (2*R)/(n(i)+1);
    % mesh nodes
    x = linspace(-R,R,n(i)+2)';
    % number of time steps
    M=ceil(T/h);
    % time step
    k=T/M;
    %----------------------------------------------------------------------
    %  Compute stiffness matrix and load vector
    %----------------------------------------------------------------------

    e = ones(n(i)+2,1);
    % mass matrix
    Am = (h/6)*spdiags([e 4*e e],-1:1,n(i)+2,n(i)+2);
    % cross matrix
    Ac = (1/2)*spdiags([-e 0*e e],-1:1,n(i)+2,n(i)+2);
    As = (1/h)*spdiags([-e 2*e -e],-1:1,n(i)+2,n(i)+2);

    % stiffness martix
    A = ((sigma^2)/2)*As + ((sigma^2)/2-r)*Ac + r*Am;


    %----------------------------------------------------------------------
    %  Solver
    %----------------------------------------------------------------------

    B = Am+k*theta*A;
    C = Am-k*(1-theta)*A;

    % homogeneous dirichlet data
    u = zeros(n(i)+2,M+1);
    % degree of freedoms
    dof = 2:n(i)+1;
```

Please turn over.

```
    u(:,1) = payoff_HANDLE(x);

    for m = 1:M
        u(dof,m+1) = B(dof,dof)\(C(dof,dof)*u(dof,m));
    end


    %-----------------------------------------------------------------------
    %  Error
    %-----------------------------------------------------------------------
    %domain of interest
    S = exp(x);
    % used to compare with log--moneyness
    I = (abs(x - log(K1)) < 1.5);

    % time grid to compute exact solutiuon at t
    t = T*(linspace(0,1,M+1));


    diff = u(I,:)-SOL_HANDLE(t,x(I));
    diff = sqrt(h*sum(diff(2:end,:).^2));

    %L_2-L_2-error
    errorL2L2(i) = sqrt(sum(k.*diff(2:end).^2));


end


pL2 = polyfit(log(n),log(errorL2L2),1);
fprintf('Price: Convergence rate in L2L2 s = %2.1f\n',pL2(1));

fig1 = figure(1);
loglog(n,errorL2L2, '-x');
hold on;
loglog(n,exp(pL2(2))*n.^pL2(1),'--')
loglog(n,1.5*exp(pL2(2))*n.^-2,'k-')
grid on
xlabel('log number of mesh points')
ylabel('log error')
str_fit = sprintf('fit: O(N^{%2.1f} )',pL2(1));
legend('L2L2 FE Price error',str_fit,'O(N^{-1.5})')



fig2 = figure(2);
plot(S(I),u(I,end),'-rx'); hold on
plot(S(I),SOL_HANDLE(T,x(I)),'-bo'); hold on
plot(S(I),payoff_HANDLE(x(I)),'-k'); hold on
legend('FE Price','Exact Price','Payoff','Location','Best')
xlabel('S')
ylabel('Option Price')

%--------------------------------------
% Save the plot (do not change)
saveas(fig1, 'L2L2error.eps', 'eps')
saveas(fig2, 'price_eu.eps', 'eps')
%--------------------------------------


%-----------------------------------------------------------------------
%
% AMERICAN OPTION WITH BUTTERFLY PAYOFF
%
```

Continued overleaf.

```
%--------------------------------------------------------------------------

%--------------------------------------------------------------------------
%  Discretization
%--------------------------------------------------------------------------

% take discretization from above
% (e.g. the mass matrix, stiffness matrix, and other parameters)
% and set
n=n(end);

% stock prize
S = exp(x);


% payoff_HANDLE = @(x) max(0, exp(x)-K1) - 2*max(0, exp(x)-K2) + max(0, exp(x)-K3);
payoff = payoff_HANDLE(x);
 % solution vector without excess to payoff
u = zeros(n+2,1);
% solution vector with excess to payoff
u2 = zeros(n+2,1);
% free boundary starting at K0
fb1 =zeros(M+1,1); fb1(1) = K0;
% free boundary starting at K2
fb2 =zeros(M+1,1); fb2(1) = K2;


%--------------------------------------------------------------------------
%  Compute Right Hand Side
%--------------------------------------------------------------------------

% compute contribution from g^C_K0
f2 = zeros(n+2,1);
j = sum((x<=log(K0)));
f2(j) = sigma^2/2 * K0 * (x(j+1)-log(K0))/h;
f2(j+1) = sigma^2/2 * K0 * (-x(j)+log(K0))/h;
f2 = f2 + r*K0* load_vec(x,@(x) (x>log(K0)));


% compute contribution from -2*g^C_K1
j = sum((x<=log(K1)));
f2(j) = f2(j) - 2*(sigma^2/2 * K1 * (x(j+1)-log(K1))/h);
f2(j+1) = f2(j+1) - 2*(sigma^2/2 * K1 * (-x(j)+log(K1))/h);
f2 = f2 -2*( r*K1* load_vec(x,@(x) (x>log(K1))));


% compute contribution from g^C_K2
j = sum((x<=log(K2)));
f2(j) = f2(j) + sigma^2/2 * K2 * (x(j+1)-log(K2))/h;
f2(j+1) = f2(j+1) + sigma^2/2 * K2 * (-x(j)+log(K2))/h;
f2 = f2 + r*K0* load_vec(x,@(x) (x>log(K2)));


%--------------------------------------------------------------------------
%  Solver
%--------------------------------------------------------------------------

% take backward Euler timestepping
theta = 1;
B = Am+theta*k*A;
C = Am-(1-theta)*k*A;


% loop over time points
for i=1:M
```

Please turn over.

```
    % compute option price with excess to payoff
     u2(dof) = psor(B(dof,dof), k*f2(dof) + C(dof,dof)*u2(dof),u2(dof));
%     lambda = zeros(n,1);
%     [u2(dof), ~] = PDAS(B(dof,dof), k*f2(dof) + C(dof,dof)*u2(dof),...
%             zeros(n,1), u2(dof), lambda);

    % compute free boundary
    i0= find(x==log(K1));

    %compute free boundaries
    J1 = find(u2(1:i0)> 1.e-6);
    fb1(i+1) = S(J1(end));

    J2 = find(u2(i0+1:end)> 1.e-6);
    fb2(i+1) = S(i0+J2(1));

end

% add payoff
u2 = u2 + payoff;
%--------------------------------------------------------------------------
%  Postprocessing
%--------------------------------------------------------------------------


% plot option price
fig3 = figure(3);

hold on
plot(S(I),u2(I),'go-')
plot(S(I),payoff(I), 'k-');
title('American Put Option')
legend('FE approx','payoff','Location','NorthEast')


% plot free boundary
fig4 = figure(4);
plot(t,fb1,linspace(0,T,M+1),fb2)
title('Free Boundaries')
legend('fb1','fb2')
xlabel('T-t')
ylabel('spot-price')

%-------------------------------------
% Save the plot (do not change)
saveas(fig3, 'price_am.eps', 'eps')
saveas(fig4, 'exercise_bd.eps', 'eps')
%-------------------------------------
```

We observe that the convergence rate in the $L^2(J; L^2(\widetilde{G}))$-norm is less than 2, cp. Figure 1. According to the theory, the rate 2 is achieved if $v_R(0) = g^{\mathrm{BF}}(\exp(\cdot)) \in H^2(G)$. This is not the case here. Since the weak derivative of $g^{\mathrm{BF}}(\exp(\cdot))$ is piecewise constant, the second weak derivative contains *dirac functionals*, which are not in $L^2(G)$. This was shown in the exercises. Precisely, $g^{\mathrm{BF}}(\exp(\cdot)) \in H^{3/2-\epsilon}(G)$ for every $\epsilon > 0$ and $g^{\mathrm{BF}}(\exp(\cdot)) \notin H^{3/2}(G)$.

The reduced convergence rate can be improved by a graded time stepping with sufficient grading parameter or by taking a few initial time steps with step size k^2.

Continued overleaf.

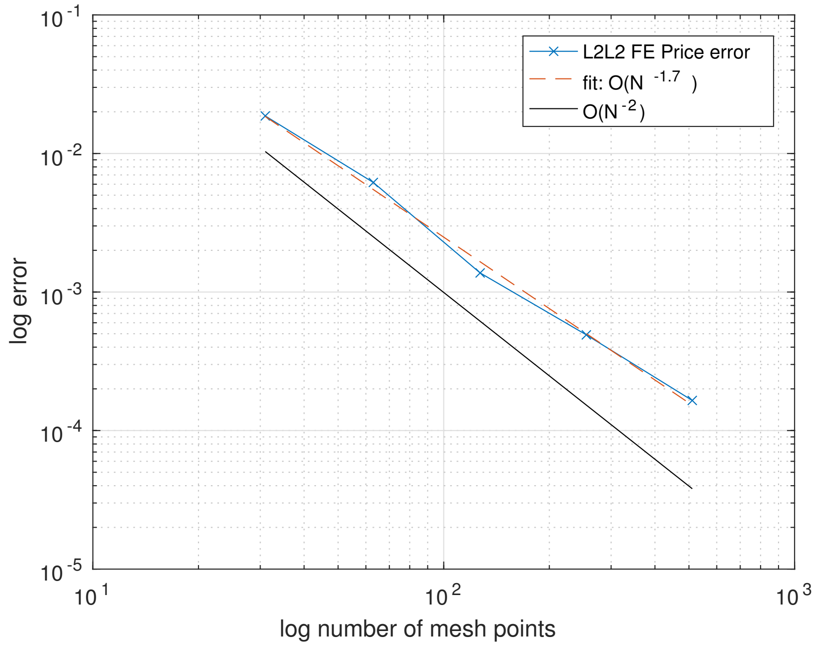

Abbildung 1: $L^2(J; L^2(\widetilde{G}))$-error.

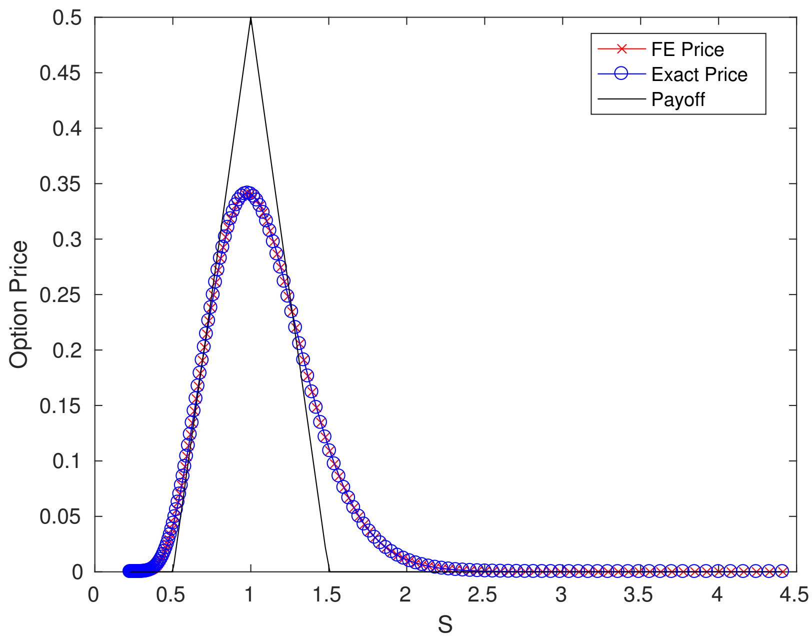

Abbildung 2: Option price and FE approximation at $t = T$ (in time-to-maturity) and payoff.

Please turn over.

e) The price of the American option with butterfly payoff is defined by

$$V^{\mathrm{Am}}(t,s) := \sup_{\tau\in\mathcal{T}_{t,T}} \mathbb{E}[e^{-r(\tau-t)} g^{\mathrm{BF}}(S_\tau)|S_t = s], \quad (t,s)\in[0,T]\times(0,\infty),$$

where $\mathcal{T}_{t,T}$ is the set of stopping times that take values in (t,T).

f) Note that $g_K^{\mathrm{C}}(e^x) = \max\{e^x - K, 0\} = (e^x - K)_+$. We obtain by direct computation

$$\begin{aligned}
-a^{\mathrm{BS}}(g_K^{\mathrm{C}}(e^x),\varphi) &= -\frac{\sigma^2}{2}\int_G g_K^{\mathrm{C}}(e^x)'\varphi'\mathrm{d}x - (\sigma^2/2 - r)\int_G g_K^{\mathrm{C}}(e^x)'\varphi\mathrm{d}x - r\int_G g_K^{\mathrm{C}}(e^x)\varphi\mathrm{d}x\\
&= -\frac{\sigma^2}{2}[e^x\varphi]_{\log(K)}^R + \frac{\sigma^2}{2}\int_{\log(K)}^R e^x\varphi\ dx - (\sigma^2/2 - r)\int_{\log(K)}^R e^x\varphi\ dx\\
&\quad - r\int_{\log(K)}^R e^x\varphi\ dx + rK\int_{\log(K)}^R \varphi\ dx\\
&= \frac{\sigma^2}{2}K\varphi(\log(K)) + rK\int_{\log(K)}^R \varphi\ dx\,.
\end{aligned}$$

For $j = 1,\ldots,N$, $i = 0,1,2$, it holds that

$$b_j(\log(K_i)) = \begin{cases} (x_{j+1} - \log(K_i))/h & \text{if } \log(K_i)\in[x_j, x_{j+1}],\\ (\log(K_i) - x_{j-1})/h & \text{if } \log(K_i)\in[x_{j-1}, x_j),\\ 0 & \text{else.}\end{cases}$$

Hence, for $j = 1,\ldots,N$,

$$\begin{aligned}
\underline{\mathrm{f}}_j &= \frac{\sigma^2}{2}b_j(\log(K_0)) + rK_0\int_G \mathbb{1}_{(\log(K_0),R)}b_j\mathrm{d}x\\
&\quad - 2\left(\frac{\sigma^2}{2}b_j(\log(K_1)) + rK_1\int_G \mathbb{1}_{(\log(K_1),R)}b_j\mathrm{d}x\right)\\
&\quad + \frac{\sigma^2}{2}b_j(\log(K_2)) + rK_2\int_G \mathbb{1}_{(\log(K_2),R)}b_j\mathrm{d}x
\end{aligned}$$

g) See the code above item d).

h) Figure 4 shows the two exercise boundaries fb_1 and fb_2. Time t is here in time-to-maturity. At $t = 0$, $\mathrm{fb}_1(0) = K_0$ and $\mathrm{fb}_2(0) = K_2$. For $t\in(0,T)$, the price satisfies for $\mathrm{fb}_1(t) < s < \mathrm{fb}_2(t)$, $V^{\mathrm{Am}}(T-t,s) = g^{\mathrm{BF}}(s)$ and for $s < \mathrm{fb}_1(t)$ and for $s > \mathrm{fb}_2(t)$, $V^{\mathrm{Am}}(T-t,s) > g^{\mathrm{BF}}(s)$. Hence, for $s\in(\mathrm{fb}_1(t),\mathrm{fb}_2(t))$, the holder should exercise the option. For $s\in(0,\mathrm{fb}_1(t))\cup(\mathrm{fb}_2(t),\infty)$, the holder should hold the option. Therefore, the *continuation region* is disconnected for butterfly payoffs

Continued overleaf.

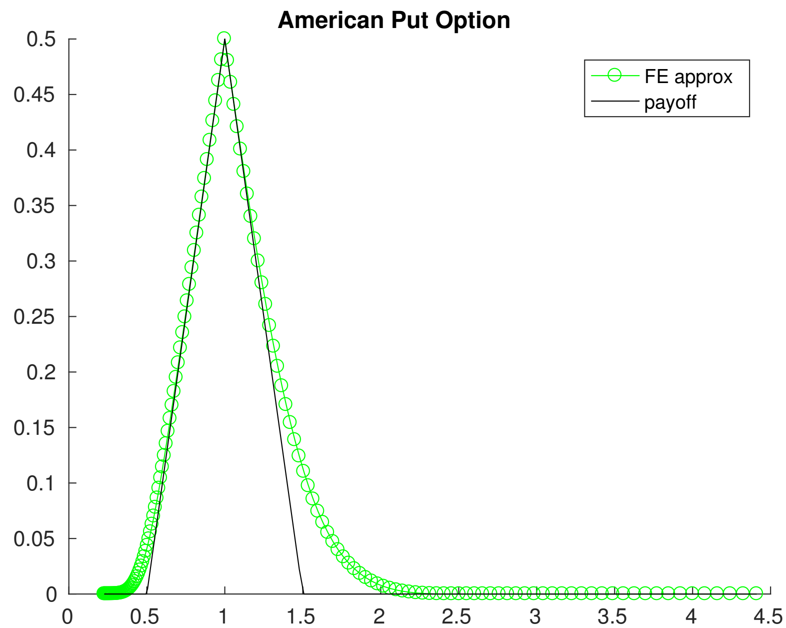

Abbildung 3: FE approximation to American option with butterfly payoff at $t = T$ (in time-to-maturity) and payoff.

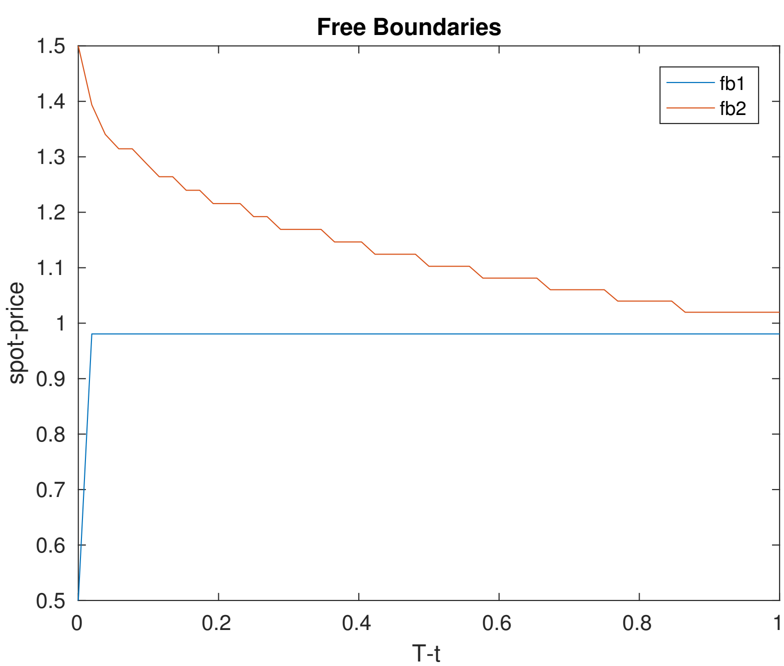

Abbildung 4: Exercise boundaries in time-to-maturity.

Please turn over.

3. **a)** ρ is the correlation coefficient and $\overline{m}$ is the long-run mean level of volatility.

b) We obtain the bilinear form $a(,\cdot,\cdot) : V \times V \to \mathbb{R}$ as follows. Let $v \in V$ be a test function. Hence, by integration by parts

$$
\begin{aligned}
a(u,v) &= \frac{1}{2}\int_{G_R} y\,\partial_x u\,\partial_x v\,\mathrm{d}x\,\mathrm{d}y + \beta\rho\int_{G_R} y\,\partial_y u\,\partial_x v\,\mathrm{d}x\,\mathrm{d}y + \frac{1}{2}\beta^2\int_{G_R} y\,\partial_y u\,\partial_y v\,\mathrm{d}x\,\mathrm{d}y \\
&\quad + \frac{1}{2}\beta^2\int_{G_R} \partial_y u\,v\,\mathrm{d}x\,\mathrm{d}y + \frac{1}{2}\int_{G_R} y\,\partial_x u\,v\,\mathrm{d}x\,\mathrm{d}y - \alpha\bar{m}\int_{G_R} \partial_y u\,v\,\mathrm{d}x\,\mathrm{d}y \\
&\quad + \alpha\int_{G_R} y\partial_y u\,v\,\mathrm{d}x\,\mathrm{d}y \\
&= \frac{1}{2}\int_{G_R} y\,\partial_x u\,\partial_x v\,\mathrm{d}x\,\mathrm{d}y + \beta\rho\int_{G_R} y\,\partial_y u\,\partial_x v\,\mathrm{d}x\,\mathrm{d}y + \frac{1}{2}\beta^2\int_{G_R} y\,\partial_y u\,\partial_y v\,\mathrm{d}x\,\mathrm{d}y \\
&\quad + \left(\frac{1}{2}\beta^2 - \alpha\bar{m}\right)\int_{G_R} \partial_y u\,v\,\mathrm{d}x\,\mathrm{d}y + \frac{1}{2}\int_{G_R} y\,\partial_x u\,v\,\mathrm{d}x\,\mathrm{d}y + \alpha\int_{G_R} y\partial_y u\,v\,\mathrm{d}x\,\mathrm{d}y\,.
\end{aligned}
$$

c) We have that $\mathbf{M} = \mathbf{M}^1 \otimes \mathbf{M}^2$ and $N = N_x(N_y+2)$, where

$$
\{\mathbf{M}^1\}_{i',i} = (b_i, b_{i'})_{L^2(-R_1,R_1)} \qquad \{\mathbf{M}^2\}_{j',j} = (b_j, b_{j'})_{L^2(0,R_2)}.
$$

The bilinear form for $u(x,y) = b_i(x)b_j(y)$, $v(x,y) = b_{i'}(x)b_{j'}(y)$ is

$$
\begin{aligned}
a(b_i(x)b_j(y), b_{i'}(x)b_{j'}(y)) &= \frac{1}{2}\int_{-R_1}^{R_1} b_i'(x)b_{i'}'(x)\mathrm{d}x \int_0^{R_2} y b_j(y)b_{j'}(y)\mathrm{d}y \\
&\quad + \beta\rho\int_{-R_1}^{R_1} b_i(x)b_{i'}'(x)\mathrm{d}x \int_0^{R_2} y b_j'(y)b_{j'}(y)\mathrm{d}y \\
&\quad + \frac{1}{2}\beta^2\int_{-R_1}^{R_1} b_i(x)b_{i'}(x)\mathrm{d}x \int_0^{R_2} y b_j'(y)b_{j'}'(y)\mathrm{d}y \\
&\quad + \left(\frac{1}{2}\beta^2 - \alpha\bar{m}\right)\int_{-R_1}^{R_1} b_i(x)b_{i'}(x)\mathrm{d}x \int_0^{R_2} b_j'(y)b_{j'}(y)\mathrm{d}y \\
&\quad + \frac{1}{2}\int_{-R_1}^{R_1} b_i'(x)b_{i'}(x)\mathrm{d}x \int_0^{R_2} y b_j(y)b_{j'}(y)\mathrm{d}y \\
&\quad + \alpha\int_{-R_1}^{R_1} b_i(x)b_{i'}(x)\mathrm{d}x \int_0^{R_2} y b_j'(y)b_{j'}(y)\mathrm{d}y\,.
\end{aligned}
$$

Therefore, the stiffness matrix has the tensor product structure

$$
\begin{aligned}
\mathbf{A} &= \frac{1}{2}\mathbf{S}^1 \otimes \mathbf{M}^y - \beta\rho\mathbf{B}^1 \otimes \mathbf{B}^y + \frac{1}{2}\beta^2\mathbf{M}^1 \otimes \mathbf{S}^y \\
&\quad + (\frac{1}{2}\beta^2 - \alpha\bar{m})\mathbf{M}^1 \otimes \mathbf{B}^2 + \frac{1}{2}\mathbf{B}^1 \otimes \mathbf{M}^y + \alpha\mathbf{M}^1 \otimes \mathbf{B}^y
\end{aligned}
$$

where the one-dimensional matrices above are given by

$$
\begin{aligned}
\{\mathbf{M}^1\}_{i',i} &= (b_i, b_{i'})_{L^2(-R_1,R_1)} & \{\mathbf{M}^y\}_{j',j} &= (y b_j, b_{j'})_{L^2(0,R_2)} \\
\{\mathbf{B}^1\}_{i',i} &= (b_i', b_{i'})_{L^2(-R_1,R_1)} & \{\mathbf{B}^y\}_{j',j} &= (y b_j', b_{j'})_{L^2(0,R_2)} \\
\{\mathbf{S}^1\}_{i',i} &= (b_i', b_{i'}')_{L^2(-R_1,R_1)} & \{\mathbf{S}^y\}_{j',j} &= (y b_j', b_{j'}')_{L^2(0,R_2)} \\
\{\mathbf{B}^2\}_{j',j} &= (b_j', b_{j'})_{L^2(0,R_2)}
\end{aligned}
$$

Continued overleaf.

d) With

$$\mathbf{Y}_{1,y} = -\beta\rho\mathbf{B}^y + \frac{1}{2}\mathbf{M}^y$$
$$\mathbf{Y}_{2,y} = \frac{1}{2}\beta^2\mathbf{S}^y + \left(\frac{1}{2}\beta^2 - \alpha\bar{m}\right)\mathbf{B}^2 + \alpha\mathbf{B}^y$$

the stiffness matrix $\mathbf{A}$ simplifies to

$$\mathbf{A} = \frac{1}{2}\mathbf{S}^1 \otimes \mathbf{M}^y + \mathbf{B}^1 \otimes \mathbf{Y}_{1,y} + \mathbf{M}^1 \otimes \mathbf{Y}_{2,y}\,,$$

e) A MATLAB implementation where the one-dimensional matrices are calculated using the routine `stiff.m` looks like this

```
%  P3_main computes European call option price with the given stochastic
%  volatility model using finite elements
%

clear all;
close all;

% -------------------------------------------------------------------------
%  Set parameters
% -------------------------------------------------------------------------

% number of nodes in x-ccordinate
Nx = 51;
% number of nodes in y-ccordinate
Ny = 51;
% number of time steps
m = 50;

% domain (-R_1,R_1)
R_1 = 4;
% domain (0,R_2)
R_2 = 3.2;

% maturity
T = 1/2;
% strike
K = 1;
rho = -0.5;
alpha = 1.5;
m_bar = 0.06;
beta = 0.7;

% -------------------------------------------------------------------------
%  Discretization
% -------------------------------------------------------------------------

% mesh size in x-coordinate
hx = (2*R_1)/(Nx+1);
% mesh size in y-coordinate
hy = (R_2)/(Ny+1);
% mesh nodes in x-coordinate
x = linspace(-R_1,R_1,Nx+2)';
% mesh nodes in y-coordinate
y = linspace(0,R_2,Ny+2)';
% time steps
k = T/m;
```

Please turn over.

```
% -------------------------------------------------------------------------
%  Compute Generator/ Source Term/ Initial Data
% -------------------------------------------------------------------------

% non-weighted matrices
e = ones(Nx,1);
M1 = hx/6*spdiags([e 4*e e],-1:1,Nx,Nx);
B1 = 1/2*spdiags([-e zeros(Nx,1) e],-1:1,Nx,Nx);
S1 = 1/hx*spdiags([-e 2*e -e],-1:1,Nx,Nx);

e = ones(Ny+2,1);
M2 = hy/6*spdiags([e 4*e e],-1:1,Ny+2,Ny+2);
B2 = 1/2*spdiags([-e zeros(Ny+2,1) e],-1:1,Ny+2,Ny+2);

% incoorporate hom. Neumann boundary conditions
M2(1,1) = hy/3; M2(Ny+2,Ny+2) = hy/3;
B2(1,1) = -1/2; B2(Ny+2,Ny+2) = 1/2;

% weighted matrices
Sy = stiff(y,@(z) z,@(z) 0, @(z) 0);
My = stiff(y,@(z) 0,@(z) 0, @(z) z);
By = stiff(y,@(z) 0,@(z) z, @(z) 0);


% define matrices Y1, Y2
Y1 = -rho*beta*By+1/2*My;
Y2 = 1/2*beta^2*Sy+(1/2*beta^2-alpha*m_bar)*B2+alpha*By;

% tensor product
M = kron(M1,M2);
A = 1/2*kron(S1,My)+kron(B1,Y1)+kron(M1,Y2);

% initial data
u0x = max(0,exp(x(2:end-1))-K); u0y = ones(length(y),1);
u0 = kron(u0x,u0y);

% -------------------------------------------------------------------------
%  Solver
% -------------------------------------------------------------------------

theta = 0.5;
B = M+k*theta*A; C = M-(1-theta)*k*A;

% lopp over time points
u = u0;
for i = 0:m-1
    u = B\(C*u);
end

% -------------------------------------------------------------------------
%  Postprocessing
% -------------------------------------------------------------------------

% area of interest
idxd = find(x <= -1,1,'last');
idxu = find(x >= 1,1);
idyd = find(y <= 0.1,1,'last');
idyu = find(y >= 1.2,1);

% compute exact solution
S = exp(x(idxd:idxu)); y = y(idyd:idyu);
uex = stochvol_exact(S,y,T,K,rho,alpha,m_bar,beta,0);

% plot option price
```

Continued overleaf.

```
u = reshape(u,Ny+2,Nx);
u = [zeros(Ny+2,1),u,zeros(Ny+2,1)];
u = u(idyd:idyu,idxd:idxu);
u0 = reshape(u0,Ny+2,Nx);
u0 = [zeros(Ny+2,1),u0,zeros(Ny+2,1)];
u0 = u0(idyd:idyu,idxd:idxu);
[X,Y] = meshgrid(S,y);

fig1 = figure(1);
surf(X,Y,u),
hold on
mesh(X,Y,u0)
title('European Call option in stoch. vol. model')
xlabel('S'), ylabel('y'), zlabel('u')

% plot error
fig2 = figure(2);
mesh(exp(X),Y,abs(u-uex))
xlabel('S'), ylabel('y'), zlabel('|e|')


%-------------------------------------
% Save the plot (do not change)
saveas(fig1, 'price_stochvol.eps', 'eps')
saveas(fig2, 'error_stochvol.eps', 'eps')
%-------------------------------------
```

The results are shown in Figure 5.

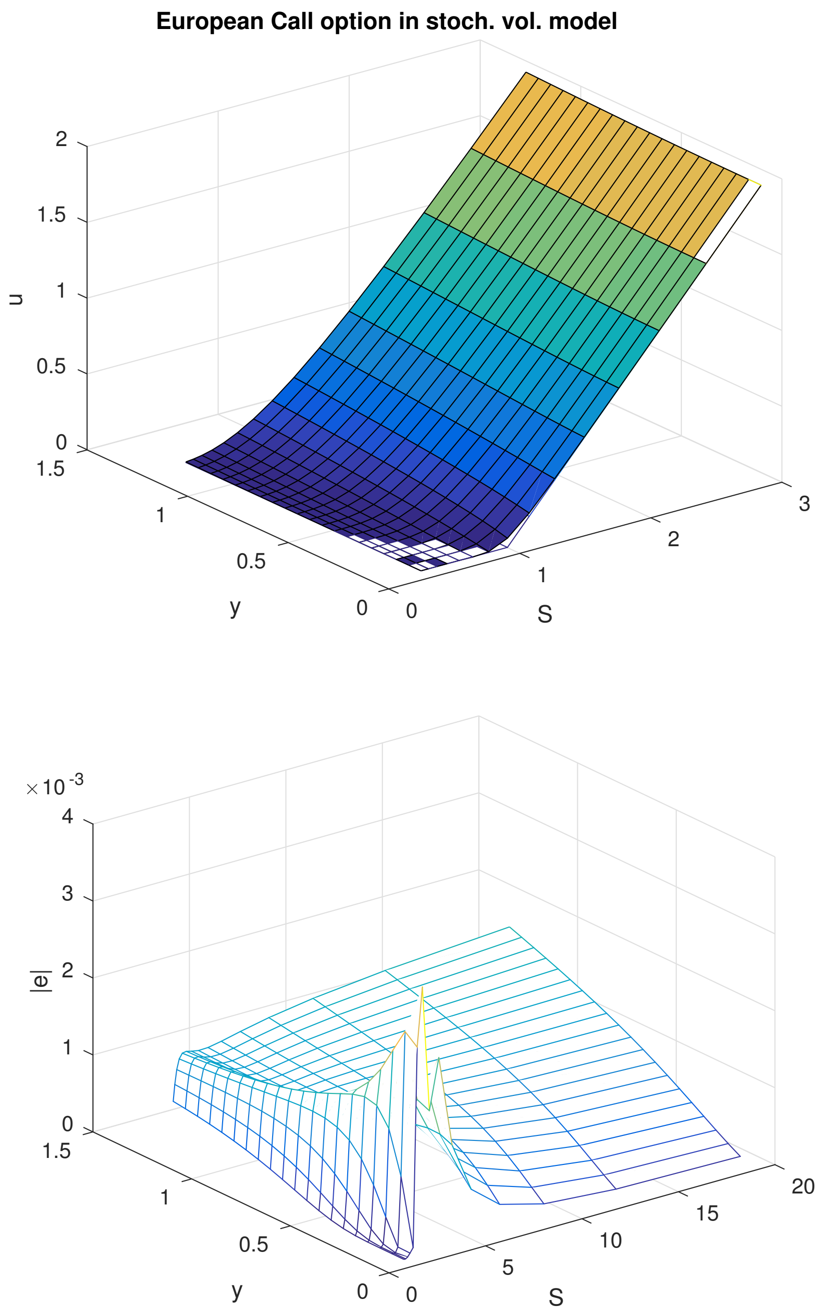

Abbildung 5: Numerical solution of the call price and the corresponding maximum error.