UNIVERSITÄT
KOBLENZ · LANDAU

MLGSA PROJECT
KO – KL

Prepared by:

Prof. Dr. Michael Hinze
M.Sc Rajit Sanghvi

Newton multigrid method for solving parabolic optimal control problems

UNIVERSITÄT KOBLENZ · LANDAU

NEWTON'S METHOD - OVERVIEW

$$Minimize\ J(y,u) \quad s.t \quad A(u,\theta) = R$$

$$L := J \ - p(A(u,\theta) - \mathrm{R})$$

Optimality conditions:

$$L_p : A(u,\theta) = R$$

$$L_u : A_u(u,\theta)p = J_u(u,\theta)$$

$$L_\theta : J_\theta(u,\theta)p = A_\theta(u,\theta)p$$

$$G \ := \begin{bmatrix} A(u,\theta) - R \\ A_u(u,\theta)p - J_u(u,\theta) \\ J_\theta(u,\theta)p - A_\theta(u,\theta)p \end{bmatrix} = 0$$

Newton step will take the following form,

$$G' \begin{bmatrix} \delta u \\ \delta p \\ \delta \theta \end{bmatrix} = -G$$

Final update,

$$\begin{pmatrix} y \\ p \\ \theta \end{pmatrix}_{ijm} = \begin{pmatrix} y \\ p \\ \theta \end{pmatrix}_{ijm} + \begin{bmatrix} \delta y \\ \delta p \\ \delta \theta \end{bmatrix}$$

Consider the following optimal control problem (P)

$$Minimize\ J(y,u)$$

$$\begin{aligned} -\partial_t y + \sigma\Delta y &= u && \text{in } Q = \Omega\, x\, (0,T) \\ y(x,0) &= y^0 && \text{in } \Omega \\ y(x,t) &= 0 && on\ \Sigma = \Gamma \text{ x } (0,\mathrm{T}) \end{aligned}$$

$$J(y,u) := \frac{\alpha}{2}||y - z||^2 + \frac{\beta}{2}||y(\cdot,T) - z(\cdot,T)||^2 + \frac{v}{2}||u||^2$$

Optimality system

$$Aw = f \quad \left\{ \begin{aligned} & -\partial_t y + \sigma\Delta y = u && y(x,0) = y^0 \\ & \partial_t p + \sigma\Delta p + \alpha(y - z) = 0 && p(x,T) = \beta\big(y(x,T) - z(x,T)\big) \\ & vu - p = 0 \end{aligned} \right.$$

$$G := f - Aw = 0$$

UNIVERSITÄT KOBLENZ · LANDAU

The multigrid and the Newton solver

The Newton algorithm in space and time can be written in defect correction form as follows:

$$w_{i+1} := w_i + M^{-1}(f - Aw_i);$$

with ***M* being the Fréchet derivative or Jacobian matrix of the operator *A*.**

$$A = \begin{bmatrix} A_0 & C_0 & & & & & & & \\ B_1 & A_1 & C_1 & & & & & & \\ & B_2 & A_2 & C_2 & & & & & \\ & & B_3 & A_3 & C_3 & & & & \\ & & & & \cdot & & & & \\ & & & & & \cdot & & & \\ & & & & & & \cdot & & \\ & & & & & & & \cdot & C_{N_{t-1}} \\ & & & & & & & B_{N_t} & A_{N_t} \end{bmatrix} \qquad w = \begin{bmatrix} y_0 \\ p_0 \\ y_1 \\ p_1 \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ y_{N_t} \\ p_{N_t} \end{bmatrix} \qquad f = \begin{bmatrix} \left(-\frac{I}{\Delta t} + \sigma\Delta\right) y^0 \\ 0 \\ 0 \\ z_1 \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ 0 \\ \beta z_{N_t} \end{bmatrix}$$

$$A_m = \begin{bmatrix} \left(-\frac{I}{\Delta t} + \sigma\Delta\right) & -\frac{1}{\upsilon} \\ \alpha & \left(-\frac{I}{\Delta t} + \sigma\Delta\right) \end{bmatrix} \qquad B_m = \begin{bmatrix} \frac{1}{\Delta t} & 0 \\ 0 & 0 \end{bmatrix} \qquad C_m = \begin{bmatrix} 0 & 0 \\ 0 & \frac{1}{\Delta t} \end{bmatrix} \qquad A_{N_t} = \begin{bmatrix} \left(-\frac{I}{\Delta t} + \sigma\Delta\right) & -\frac{1}{\upsilon} \\ \beta & 1 \end{bmatrix}$$

Solving Optimality system, discretization of optimality system by finite difference and backward Euler scheme

State equation:

$$-\left(\frac{y_{ijm} - y_{ijm-1}}{\delta t}\right) + \sigma\left(\frac{y_{i+1jm} + y_{i-1jm} + y_{ij-1m} + y_{ij+1m} - 4y_{ijm}}{\delta t}\right) - \frac{p_{ijm}}{\upsilon} = 0$$

$$-[1+4\sigma]y_{ijm} + \sigma\left(y_{i+1jm} + y_{i-1jm} + y_{ij-1m} + y_{ij+1m} - 4y_{ijm}\right) + y_{ijm-1} - \frac{\delta t\, p_{ijm}}{\upsilon} = 0$$

Adjoint equation: $$\left(\frac{p_{ijm+1} - p_{ijm}}{\delta t}\right) + \sigma\left(\frac{p_{i+1jm} + p_{i-1jm} + p_{ij-1m} + p_{ij+1m} - 4p_{ijm}}{\delta t}\right) + \alpha\left(y_{ijm} - z_{ijm}\right) = 0$$

$$-[1+4\sigma]p_{ijm} + \sigma\left(p_{i+1jm} + p_{i-1jm} + p_{ij-1m} + p_{ij+1m} - 4p_{ijm}\right) + p_{ijm+1} + \delta t\alpha\left(y_{ijm} - z_{ijm}\right) = 0$$

UNIVERSITÄT KOBLENZ · LANDAU

Collective smoothing step:

$$\begin{pmatrix} y \\ p \end{pmatrix}_{ijm} = \begin{pmatrix} y \\ p \end{pmatrix}_{ijm} + M_{ijm}^{-1} \begin{pmatrix} r_y \\ r_p \end{pmatrix}_{ijm}$$

Where,

$$M_{ijm} = \begin{bmatrix} -(1+4\sigma) & -\dfrac{\delta t}{v} \\ \delta t\ \alpha & -(1+4\sigma) \end{bmatrix}_{ijm}$$

$$\begin{pmatrix} y \\ p \end{pmatrix}_{ijm} = \begin{pmatrix} y \\ p \end{pmatrix}_{ijm} + \begin{bmatrix} -(1+4\sigma) & -\frac{\delta t}{v} \\ \delta t\ \alpha & -(1+4\sigma) \end{bmatrix}_{ijm}^{-1} \begin{pmatrix} r_y \\ r_p \end{pmatrix}_{ijm}$$

UNIVERSITÄT KOBLENZ · LANDAU

Considering the discrete optimality systems at any i, j and for all time steps, a **block – tridiagonal system** is obtained

$$W = (\delta y_0, \delta p_0, \delta y_1, \delta p_0, \ldots, \delta y_N, \delta p_N)$$

$$M\, w_i = r_i$$

$$M = \begin{bmatrix} A_0 & C_0 & & & & & & & \\ B_1 & A_1 & C_1 & & & & & & \\ & B_2 & A_2 & C_2 & & & & & \\ & & B_3 & A_3 & C_3 & & & & \\ & & & & . & & & & \\ & & & & & . & & & \\ & & & & & & . & & \\ & & & & & & & . & C_{N_{t-1}} \\ & & & & & & & B_{N_t} & A_{N_t} \end{bmatrix} \qquad w = \begin{bmatrix} \delta y_0 \\ \delta p_0 \\ \delta y_1 \\ \delta p_1 \\ . \\ . \\ . \\ . \\ \delta y_{N_t} \\ \delta p_{N_t} \end{bmatrix}$$

$$A_m = \begin{bmatrix} -(1+4\sigma) & -\frac{\delta t}{\upsilon} \\ \delta t\ \alpha & -(1+4\sigma) \end{bmatrix} \qquad B_m = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix} \qquad C_m = \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix} \qquad A_{N_t} = \begin{bmatrix} -(1+4\sigma) & -\frac{\delta t}{\upsilon} \\ \beta & -1 \end{bmatrix}$$

$$\beta(y_h^m - z_h^m) - p_h^m = 0$$

The multigrid solver to solve error equation

$$M\, w_i = r_i$$

Algorithm : Space-time multigrid

Function $SpaceTimeMultigrid(w, r, k)$
If $(k = 1)$ **then**
return $M^{-1}r$
End if
While (not converged) **do**
$w \leftarrow S\left(M^k, w, r, NSMpre\right)$ *presmoothing*
$d \leftarrow R\left(r - M^k w\right)$ restriction.
$w \leftarrow w + I($SpaceTimeMultigrid$(0; d; k-1))$ prolongation
$w \leftarrow S\left(M^k, w, r, NSMpost\right)$ *postsmoothing*
end while
return w
End function

UNIVERSITÄT KOBLENZ · LANDAU

SMOOTHING OPERATORS

- Classical iterative schemes are based on a fixed point iteration for solving the linear system of equations.

$$Au = f$$
$$A = M - N$$

- The Jacobi and Damped Jacobi Method, **M = diag(A) = D**

$$u^{m+1} = u^m + \omega D^{-1} r^m$$

THE GAUSS–SEIDEL METHOD AND THE SOR METHOD

- Gauss seidel.

$$\mathrm{M} = \mathrm{D} + \mathrm{L} \text{ and } \mathrm{N} = -\mathrm{U}$$

- SOR Method

$$M = \omega^{-1} D + L \text{ and } N = \omega^{-1} D - (D - U)$$

UNIVERSITÄT KOBLENZ · LANDAU

FORWARD-BACKWARD BLOCK-SOR SMOOTHER

Function FBSORSmoother(M^k, w, r, NSM)

$r \leftarrow r - M^k w$

$\mathbf{x} \leftarrow 0$

for i = 1 to *NSM* do

$\mathbf{x}^{\boldsymbol{old}} \leftarrow \mathbf{x}$

for *i* = 0 to *N* do

$d_i \leftarrow r_i - \widetilde{M}_i\left(\omega x_{i-1} + (1-\omega) x_{i-1}^{old}\right) - M_i^k x_i^{old} - \widehat{M}_i x_{i+1}^{old}$ Forward in time

$x_i \leftarrow x_i^{old} + (M_i^k)^{-1} d_i$

end for

$\mathbf{x}^{\boldsymbol{old}} \leftarrow \mathbf{x}$

for *i* = *N* downto 0 do

$d_i \leftarrow r_i - \widetilde{M}_i x_{i-1}^{old} - M_i^k x_i^{old} - \widehat{M}_i\left(\omega x_{i+1} + (1-\omega) x_{i+1}^{old}\right)$ Backward in time

$x_i \leftarrow x_i^{old} + (M_i^k)^{-1} d_i$

end for

end for

w $\leftarrow w + x$

return w

End function

2. Model problem 2 $(\boldsymbol{P_{con}})$

$$Minimize\ J(y,u)$$

$$\partial_t y - y'' + u\,y = 0 \qquad \text{in } Q = \Omega \times (0,T)$$
$$y(x,0) = y^0 \qquad \text{in } \Omega$$
$$y(x,t) = 0 \qquad on\ \Sigma = \Gamma \times (0,T)$$

$$J(y,u) := \frac{1}{2}||y - z||^2 + \frac{v}{2}||u||^2$$

Optimality system in terms of projections

$$\partial_t y - y'' + uy = 0 \qquad y(x,0) = y^0$$
$$-\partial_t p - p'' + up = (y - z) \qquad p(x,T) = 0$$
$$vu - py = 0$$

and conditions

$$u \in U_{ad} := \{u \in L^2(Q):\ u_a \leq u \leq u_b\ a.e\ in\ Q\},$$

$$u = P_{[a,b]}\left\{-\frac{py}{v}\right\}$$

1. Model problem 1 (P)

$$Minimize\ J(y,u)$$

$$\partial_t y - y'' = u \qquad \text{in } Q = \Omega \text{ x } (0,T)$$
$$y(x,0) = y^0 \qquad \text{in } \Omega$$
$$y(x,t) = 0 \qquad on\ \Sigma = \Gamma \text{ x } (0,\mathrm{T})$$

$$J(y,u) := \frac{1}{2}||y - z||^2 + \frac{\beta}{2}||y(\cdot,\mathrm{T}) - z(\cdot,T)||^2 + \frac{v}{2}||u||^2$$

Optimality system

$$Aw = f \quad \begin{cases} \partial_t y - y'' = u & y(x,0) = y^0 \\ -\partial_t p - p'' = (y - z) & p(x,T) = \beta(y(\cdot,\mathrm{T}) - z(\cdot,T)) \\ vu + p = 0 & \end{cases}$$

$$G := f - Aw = 0$$

UNIVERSITÄT KOBLENZ · LANDAU

Solving Optimality system, discretization of optimality system by finite difference and backward Euler scheme

State equation:

$$\boxed{\left(\frac{y_{im+1} - y_{im}}{\delta t}\right) - D \; y_{im+1} = u_{im+1}} \qquad 0 \leq m \leq N-1$$

$$\left(\frac{1}{\delta t} - D\right) y_{im+1} - \frac{1}{\delta t} y_{im} = u_{im+1}$$

$$\begin{bmatrix} \left(\frac{1}{\delta t} - D\right) & & & \\ -\frac{1}{\delta t} & \left(\frac{1}{\delta t} - D\right) & & \\ & . & . & \\ & & -\frac{1}{\delta t} & \left(\frac{1}{\delta t} - D\right) \end{bmatrix} \begin{bmatrix} y_{i0} \\ y_{i1} \\ y_{i2} \\ . \\ y_{iN} \end{bmatrix} = \begin{bmatrix} \left(\frac{1}{\delta t} - D\right) y^0 \\ u_{i1} \\ u_{i2} \\ . \\ u_{iN} \end{bmatrix}$$

Solving Optimality system, discretization of optimality system by finite difference and backward Euler scheme

Adjoint equation: $-\partial_t p - p'' = (y - z)$

$$\begin{bmatrix} \left(\frac{1}{\delta t} - D\right) & -\frac{1}{\delta t} & & \\ & \left(\frac{1}{\delta t} - D\right) & . & \\ & & . & -\frac{1}{\delta t} \\ & & & \left(\frac{1}{\delta t} - D\right) \end{bmatrix} \begin{bmatrix} p_0 \\ p_1 \\ p_2 \\ . \\ p_N \end{bmatrix} = \begin{bmatrix} y_0 - z_0 \\ y_1 - z_1 \\ y_2 - z_2 \\ . \\ \beta(y_N - z_N) \end{bmatrix}$$

$$\left(\frac{1}{\delta t} - D\right) p_0 - \frac{1}{\delta t} p_1 = y_0 - z_0$$

$$\left(\frac{1}{\delta t} - D\right) p_{N-1} - \frac{1}{\delta t} p_N = y_{N-1} - z_{N-1}$$

$$\boxed{-\left(\frac{p_{im} - p_{im-1}}{\delta t}\right) - D\ p_{im-1} = y_{im-1} - z_{im-1}} \qquad 1 \leq m \leq N$$

$$A = \begin{bmatrix} \left(\frac{1}{\delta t} - D\right) & & & \\ -\frac{1}{\delta t} & \left(\frac{1}{\delta t} - D\right) & & \\ & \ddots & \ddots & \\ & & -\frac{1}{\delta t} & \left(\frac{1}{\delta t} - D\right) \end{bmatrix}$$

$$A^T = \begin{bmatrix} \left(\frac{1}{\delta t} - D\right) & -\frac{1}{\delta t} & & \\ & \left(\frac{1}{\delta t} - D\right) & \ddots & \\ & & \ddots & -\frac{1}{\delta t} \\ & & & \left(\frac{1}{\delta t} - D\right) \end{bmatrix}$$

$$AA^T = \begin{bmatrix} \left(\frac{1}{\delta t} - D\right)^2 & -\frac{1}{\delta t}\left(\frac{1}{\delta t} - D\right) & & \\ -\frac{1}{\delta t}\left(\frac{1}{\delta t} - D\right) & \left(\frac{1}{\delta t} - D\right)^2 + \left(\frac{1}{\delta t}\right)^2 & & \\ & & \ddots & -\frac{1}{\delta t}\left(\frac{1}{\delta t} - D\right) \\ & \ddots & -\frac{1}{\delta t}\left(\frac{1}{\delta t} - D\right) & \left(\frac{1}{\delta t} - D\right)^2 + \left(\frac{1}{\delta t}\right)^2 \end{bmatrix}$$

Solving Optimality system, discretization of optimality system by finite difference and backward Euler scheme

State equation:

$$\left(\frac{y_{im+1} - y_{im}}{\delta t}\right) - D\ y_{im+1} + \frac{p_{im+1}}{\upsilon} = 0$$

$$\left(\frac{I}{\delta t} - D\right) y_{im+1} - \frac{I}{\delta t}\, y_{im} + \frac{p_{im+1}}{\upsilon} = 0$$

Adjoint equation:

$$-\left(\frac{p_{im} - p_{im-1}}{\delta t}\right) - D\ p_{im-1} = y_{im-1} - z_{im-1}$$

$$\left(\frac{I}{\delta t} - D\right) p_{im-1} - \frac{I}{\delta t}\ p_{im} = y_{im-1} - z_{im-1}$$

$$A = \begin{bmatrix} A_0 & C_0 & & & & & & & \\ B_1 & A_1 & C_1 & & & & & & \\ & B_2 & A_2 & C_2 & & & & & \\ & & B_3 & A_3 & C_3 & & & & \\ & & & & . & & & & \\ & & & & & . & & & \\ & & & & & & . & & \\ & & & & & & & . & C_{N_{t-1}} \\ & & & & & & & B_{N_t} & A_{N_t} \end{bmatrix} \qquad w = \begin{bmatrix} y_{i0} \\ p_{i0} \\ y_{i1} \\ p_{i1} \\ . \\ . \\ . \\ . \\ y_{iN_t} \\ p_{iN_t} \end{bmatrix} \qquad f = \begin{bmatrix} y^0 \\ -z_0 \\ 0 \\ -z_1 \\ . \\ . \\ . \\ . \\ 0 \\ -\beta z_{N_t} \end{bmatrix}$$

$$A_0 = \begin{bmatrix} 1_{ixi} & 0 \\ -1_{ixi} & \left(\frac{I}{\delta t} - D\right)_{ixi} \end{bmatrix} \qquad A_m = \begin{bmatrix} \left(\frac{I}{\delta t} - D\right)_{ixi} & \frac{1}{v_{ixi}} \\ -1_{ixi} & \left(\frac{I}{\delta t} - D\right)_{ixi} \end{bmatrix} \qquad A_{N_t} = \begin{bmatrix} \left(\frac{I}{\delta t} - D\right)_{ixi} & \frac{1}{v_{ixi}} \\ \beta & 1_{ixi} \end{bmatrix}$$

$$B_m = \begin{bmatrix} -\frac{1}{\delta t_{ixi}} & 0 \\ 0 & 0 \end{bmatrix} \qquad C_m = \begin{bmatrix} 0 & 0 \\ 0 & -\frac{1}{\delta t_{ixi}} \end{bmatrix}$$

UNIVERSITÄT KOBLENZ · LANDAU

Space time Smoothing scheme

State equation: $\left(\frac{y_{im+1} - y_{im}}{\delta t}\right) - D\ y_{im+1} - \frac{p_{im+1}}{v} = 0$

$$\left(-\frac{y_{im}}{\delta t}\right) - \left(\frac{y_{i+1\,m+1} - 3\,y_{i\,m+1} + y_{i-1\,m+1}}{\delta x^2}\right) - \frac{p_{i\,m+1}}{v} = 0$$

$$\left(-\frac{y_{im}}{\delta t}\right) - S_{i\,m} = 0$$

Where,

$$S_{i\,m} = \left(\frac{y_{i+1\,m+1} - 3\,y_{i\,m+1} + y_{i-1\,m+1}}{\delta x^2}\right) + \frac{p_{i\,m+1}}{v}$$

$$\left(r_y\right)_{im} = \left(-\frac{y_{im}}{\delta t}\right) - S_{i\,m}$$

Space time Smoothing scheme

Adjoint equation: $-\left(\frac{p_{im} - p_{im-1}}{\delta t}\right) - D\ p_{im-1} = y_{im-1} - z_{im-1}$

$$-\left(\frac{p_{im} - p_{im-1}}{\delta t}\right) - \left(\frac{p_{i+1\,m-1} - 2\,p_{i\,m-1} + p_{i-1\,m-1}}{\delta x^2}\right) - y_{im-1} + z_{im-1} = 0$$

$$\left(-\frac{p_{im}}{\delta t}\right) - \left(\frac{p_{i+1\,m-1} - 3\,p_{i\,m-1} + p_{i-1\,m-1}}{\delta x^2}\right) - y_{im-1} + z_{im-1} = 0$$

$$\left(-\frac{p_{im}}{\delta t}\right) - R_{i\,m} = 0$$

Where,

$$R_{i\,m} = \left(\frac{p_{i+1\,m-1} - 3\,p_{i\,m-1} + p_{i-1\,m-1}}{\delta x^2}\right) + y_{im-1} - z_{im-1}$$

$$(r_p)_{im} = \left(-\frac{p_{im}}{\delta t}\right) - R_{i\,m}$$

The multigrid solver to solve error equation

$$A\, w_i = f_i$$

Algorithm : Space-time multigrid

Function $SpaceTimeMultigrid(w, f, k)$
  If $(k = 1)$ **then**
    return $A^{-1} f$
  End if
  While (not converged) **do**
    $w \leftarrow S(A^k, w, f, NSMpre)$ *presmoothing*
    $d \leftarrow R(f - A^k w)$ restriction.
    $w \leftarrow w + I(\text{SpaceTimeMultigrid}(0; d; k-1))$ prolongation
    $w \leftarrow S(A^k, w, f, NSMpost)$ *postsmoothing*
  end while
  return w
End function

UNIVERSITÄT KOBLENZ · LANDAU

BLOCK JACOBI SMOOTHER

$$Aw = f$$

$$(L + D + U)\, w = f$$

$$D\, w = -\,(L + U)w + f$$

$$D\, w^{i+1} = -\,(L + U)w^{i} + f$$

$$D\, w^{i+1} = -\,(A\ - D)w^{i} + f$$

$$D\, w^{i+1} = Dw^{i} + (f\ - Aw^{i})$$

$$w^{i+1} = w^{i} + D^{-1}(f\ - Aw^{i})$$

UNIVERSITÄT KOBLENZ · LANDAU

BLOCK JACOBI SMOOTHER

Algorithm : Space-time Block-Jacobi smoother

Function JacSmoother$\left(A^k, w, f, NSM\right)$
    for j=0 to NSM do
        $d \leftarrow f - A^k w$
        $d \leftarrow (D^k)^{-1} d$
        $\mathrm{w} \leftarrow w + \omega d$
    end for
    return w
End function

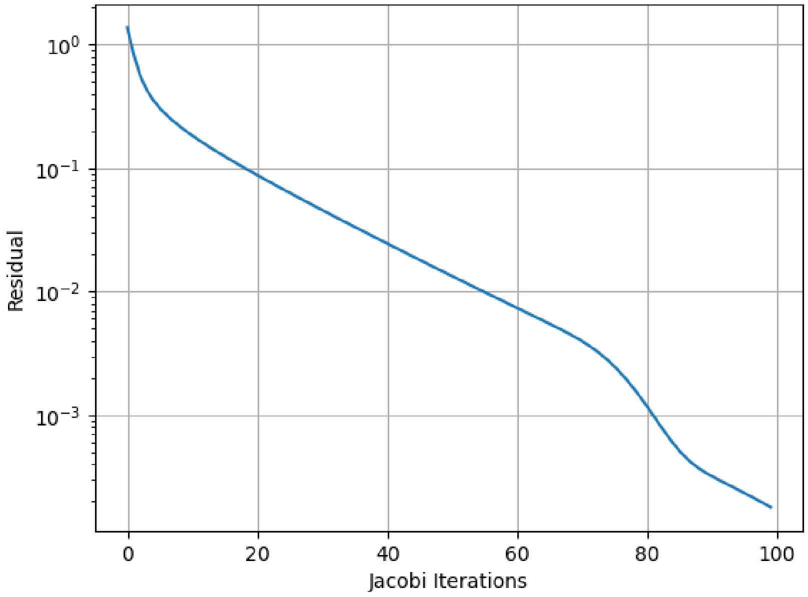
Algorithm : Space-time Block-Jacobi smoother

Function JacSmoother$\left(A^k, w, f, NSM\right)$
    for j=0 to NSM do
        $d \leftarrow f - A^k w$
        for $i \ = \ 0 \ to \ N$ do
            $d_i \leftarrow (D_i^k)^{-1} d_i$
        end for
        $\mathrm{w} \leftarrow w + \omega d$
    end for
    return w
End function

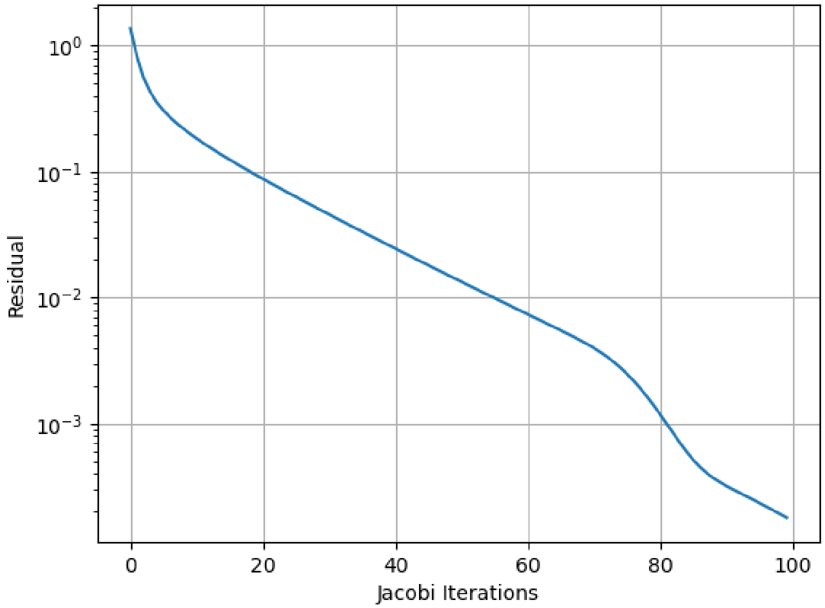
BLOCK JACOBI SMOOTHER RESIDUAL PLOT

$Time\ step: 50$ $\Delta t: 0.020408$

$Grid\ size: 5$ $\Delta x: 0.25$



$Execution\ time: \mathbf{94.919}\ \boldsymbol{Sec}$



$Execution\ time: \mathbf{0.54606}\ \boldsymbol{Sec}$

ANALYTICAL SOLUTION

$$y(x,t) = (1-t)^2 \sin(\pi x)$$

$$\partial_t y \; - y'' = u$$

$$u(x,t) = -2(1-t)\sin(\pi x) + \pi^2 (1-t)^2 \sin(\pi x)$$

$$\upsilon u + p = 0$$

$$p(x,t) = -\upsilon\,[-2(1-t)\sin(\pi x) + \pi^2(1-t)^2 \sin(\pi x)\,]$$

$$p(x,T) = 0, \qquad T = 1$$

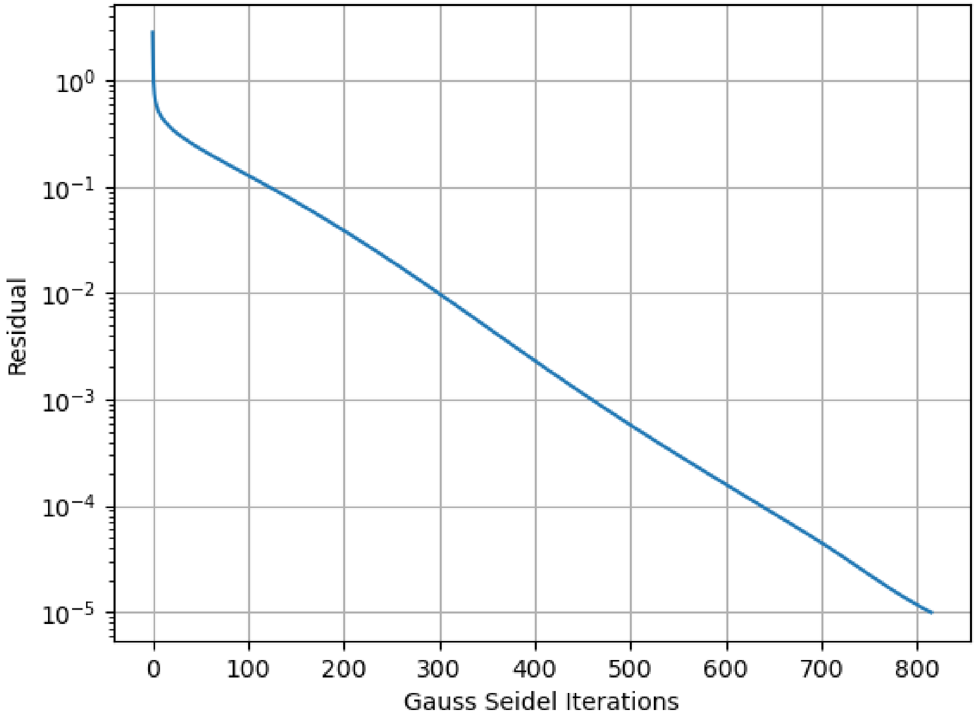
$$-\,\partial_t p \; - p'' = (y - z)$$

$$z(x,t) = (1-t)^2 \sin(\pi x)\;\; [1 + \upsilon\,\pi^4] - 2\,\upsilon \sin(\pi x)$$

BLOCK JACOBI SMOOTHER RESIDUAL PLOT

$Time\ step: 256$ $\Delta t: 0.003390$ $Tol: 1E-05$

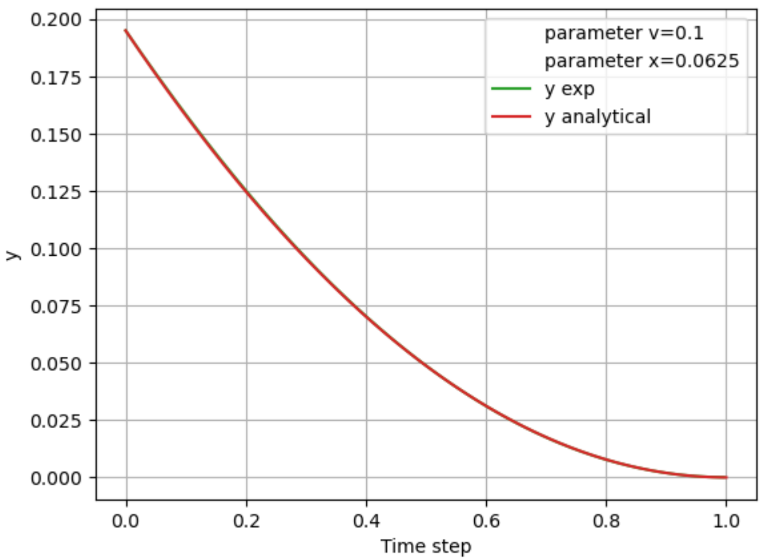
$Grid\ size: 16$ $\Delta x: 0.0625$

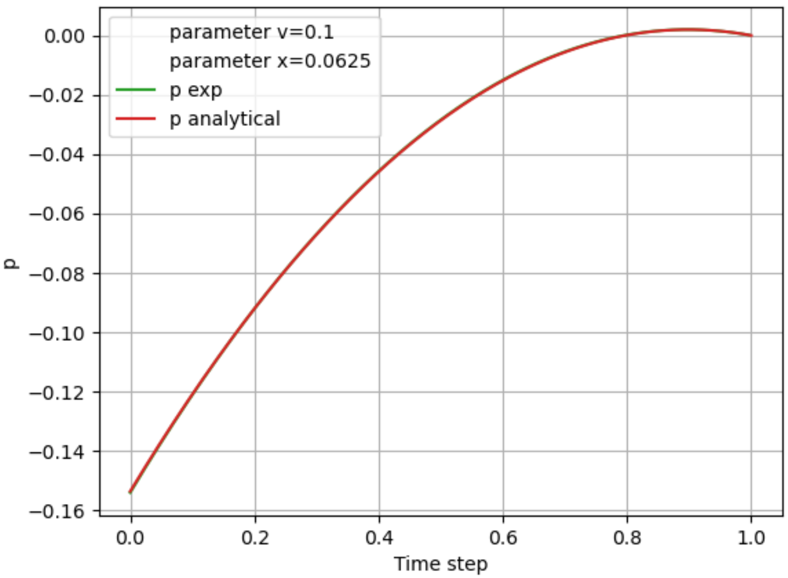


BLOCK JACOBI SMOOTHER

$$Time\ step: 256 \qquad \Delta t: 0.00390$$
$$Space\ Grid\ size: 16 \qquad \Delta x: 0.0625$$

$$\nu = 0.1$$





$$||y - y_h|| = 0.007344 \qquad ||p - p_h|| = 0.002778$$

UNIVERSITÄT KOBLENZ · LANDAU

BLOCK JACOBI SMOOTHER

$Time\ step: 256$ $\Delta t: 0.003906$

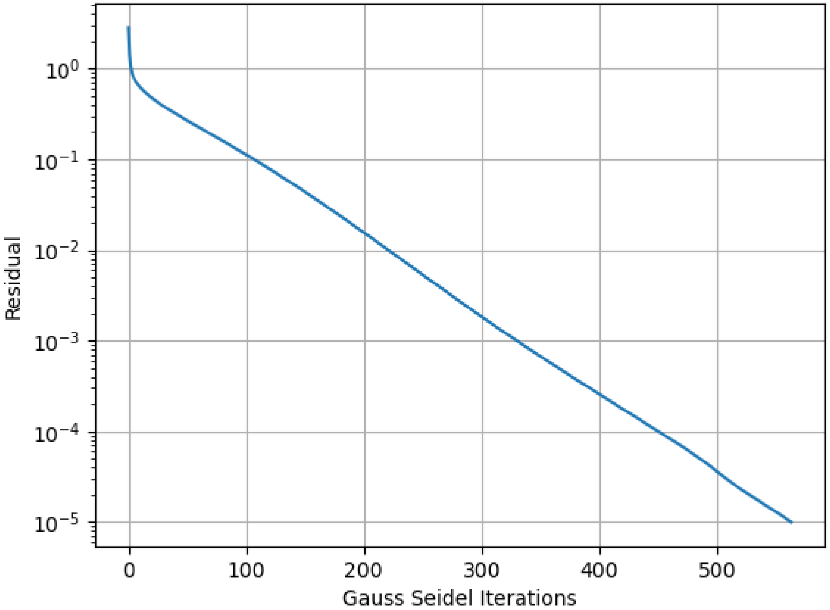
$Grid\ size: 16$ $\Delta x: 0.0625$

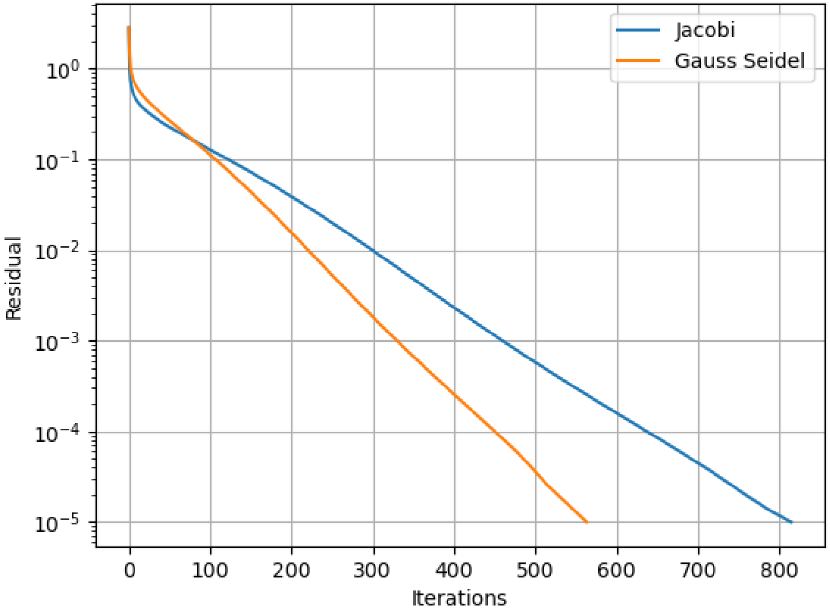
	$\|\|y - y_h\|\|$	$\|\|p - p_h\|\|$
0.0625	0.007344	0.002778
0.125	0.014423	0.005465
0.1875	0.020909	0.007910
0.25	0.026590	0.010051
0.3125	0.031288	0.011836
0.375	0.034763	0.013153
0.4375	0.036879	0.013942
0.5	0.037629	0.014239
0.5625	0.036879	0.013942
0.625	0.034763	0.013153
0.6875	0.031288	0.011836
0.75	0.026590	0.010051
0.8125	0.020909	0.007910
0.875	0.014423	0.005465
0.9375	0.007344	0.002778

BLOCK JACOBI SMOOTHER RESIDUAL PLOT

$Time\ step: 256$ $\Delta t: 0.003390$ $Tol: 1E-05$

$Grid\ size: 16$ $\Delta x: 0.0625$



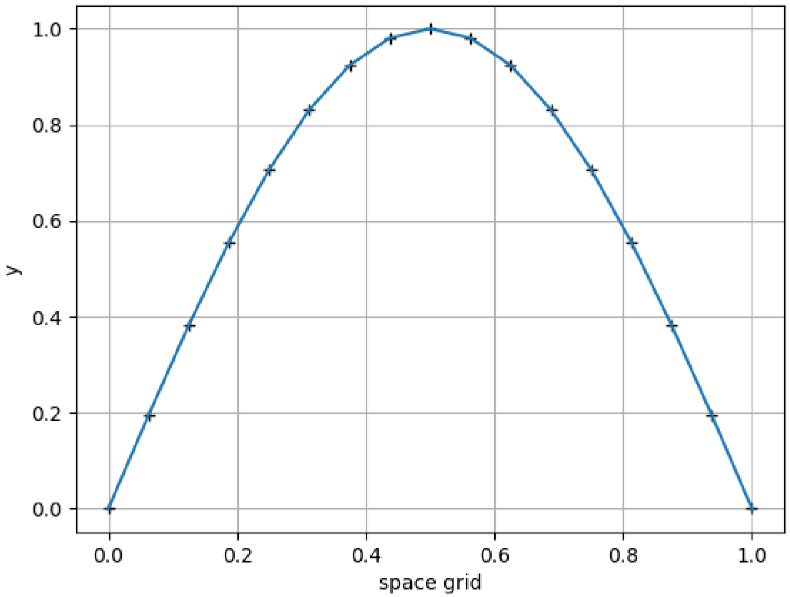


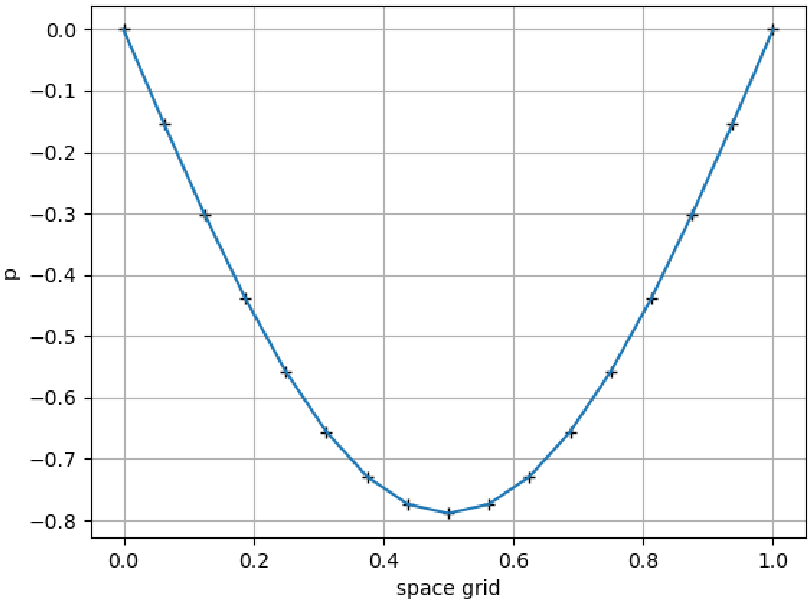
SPACE GRID VALUES FOR EACH TIME STEP

$Time\ step: 256$ $\Delta t: 0.00390$

$Space\ Grid\ size: 16$ $\Delta x: 0.0625$

$\nu = 0.1$



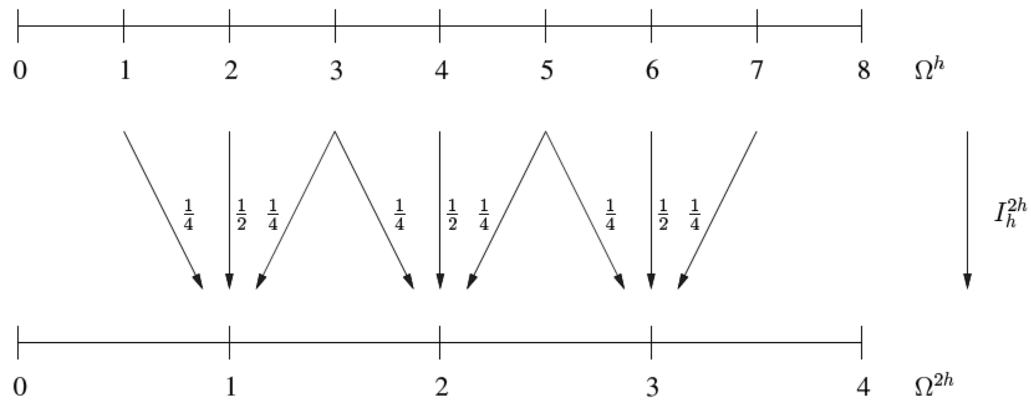


PROLONGATION/RESTRICTION

$$P\left(w^{k}\right) := \left(P_s\left(w_0^k\right), \frac{P_s\left(w_0^k\right) + P_s\left(w_1^k\right)}{2}, P_s\left(w_1^k\right), \frac{P_s\left(w_1^k\right) + P_s\left(w_2^k\right)}{2}, \dots., P_s\left(w_N^k\right)\right)$$

$$R\left(d^{k}\right) := \left(R_s\left(\frac{2d_0^k + d_1^k}{4}\right), R_s\left(\frac{d_1^k + 2d_2^k + d_3^k}{4}\right), \dots., R_s\left(\frac{d_{2N-1}^k + 2d_{2N}^k}{4}\right)\right)$$

RESTRICTION



$$R_s = \frac{1}{4}\begin{bmatrix} 1 & 2 & 1 & & & & & & & & \\ & & 1 & 2 & 1 & & & & & & \\ & & & & 1 & 2 & 1 & & & & \\ & & & & & & & \cdot & & & \\ & & & & & & & & \cdot & & \\ & & & & & & & & & 1 & 2 & 1 \end{bmatrix}$$

RESTRICTION

$$R_s v^h = \frac{1}{4}\begin{bmatrix} 1 & 2 & 1 & & & & \\ & & 1 & 2 & 1 & & \\ & & & & 1 & 2 & 1 \end{bmatrix}\begin{bmatrix} v_1^h \\ v_2^h \\ v_3^h \\ v_4^h \\ v_5^h \\ v_6^h \\ v_7^h \end{bmatrix} = \begin{bmatrix} v_1^H \\ v_2^H \\ v_3^H \end{bmatrix} = v^H$$

PROLONGATION

$$P_s = 2 * (R_s)^T$$

$$P_s = \frac{1}{2}\begin{bmatrix} 1 & & & & & \\ 2 & & & & & \\ 1 & 1 & & & & \\ & 2 & & & & \\ & 1 & 1 & & & \\ & & 2 & & & \\ & & 1 & & & \\ & & & . & & \\ & & & & . & \\ & & & & & 1 \\ & & & & & 2 \\ & & & & & 1 \end{bmatrix}$$

PROLONGATION

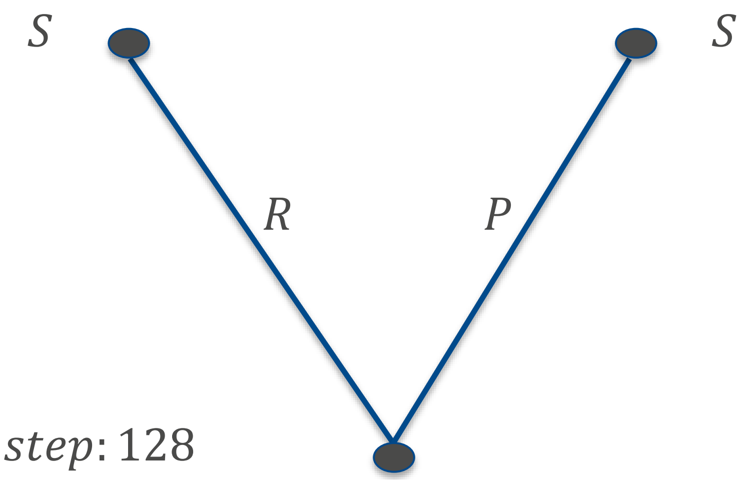
$$P_s\, v^H = \frac{1}{2}\begin{bmatrix} 1 & & \\ 2 & & \\ 1 & 1 & \\ & 2 & \\ & 1 & 1 \\ & & 2 \\ & & 1 \end{bmatrix}\begin{bmatrix} v_1^H \\ v_2^H \\ v_3^H \end{bmatrix} = \begin{bmatrix} v_1^h \\ v_2^h \\ v_3^h \\ v_4^h \\ v_5^h \\ v_6^h \\ v_7^h \end{bmatrix} = v^h$$

UNIVERSITÄT KOBLENZ · LANDAU

MULTIGRID V CYCLE

$$A\,w = f$$

$Time\ step: 256$
$Space\ Grid\ size: 16$



$Time\ step: 128$
$Space\ Grid\ size: 8$

$$A^{-1}d$$

$$A\,\delta w = d, with\ \delta w_{init} = 0$$

- “small residual, large error”

COARSE GRID RESIDUAL EQUATION

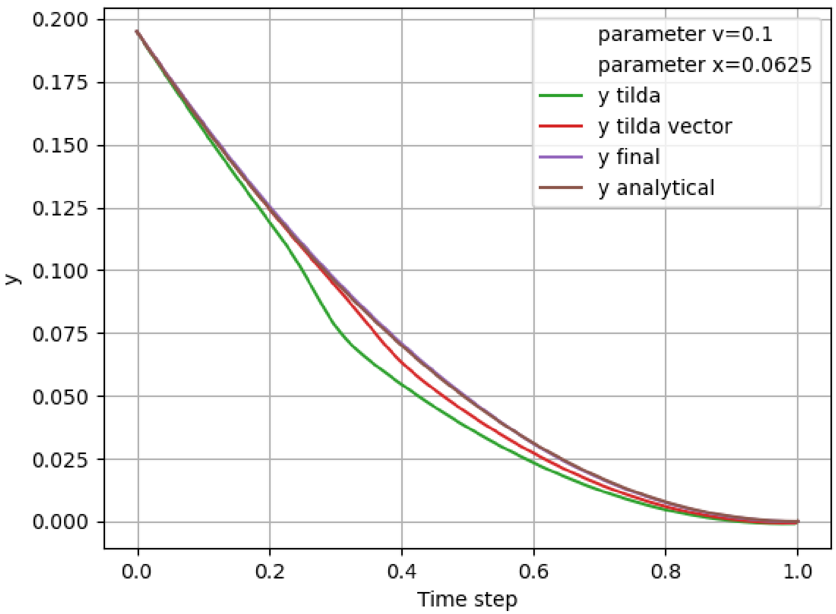
- Fine grid

$$\begin{array}{ccc} \mathit{Time\ step}: 256 & \Delta t: 0.003390 & \\ \mathit{Space\ Grid\ size}: 16 & \Delta x: 0.0625 & \Delta t = \Delta x^2 \end{array}$$

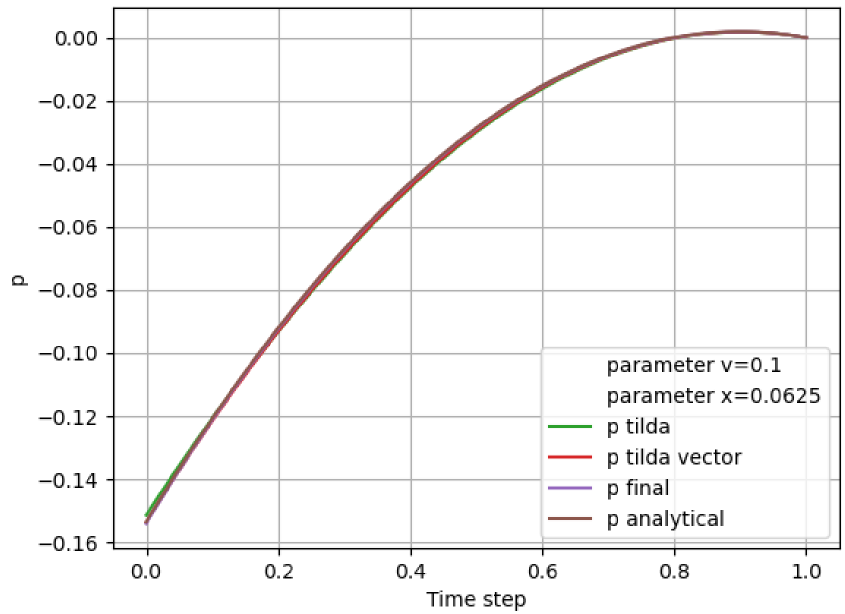
- Coarse grid

$$\begin{array}{ccc} \mathit{Time\ step}: 128 & \Delta t: 0.0078125 & \\ \mathit{Space\ Grid\ size}: 8 & \Delta x: 0.125 & \Delta t \neq \Delta x^2 \end{array}$$

RESULTS



parameter v=0.1
parameter x=0.0625
y tilda
y tilda vector
y final
y analytical
y
Time step
0.200
0.175
0.150
0.125
0.100
0.075
0.050
0.025
0.000
0.0
0.2
0.4
0.6
0.8
1.0

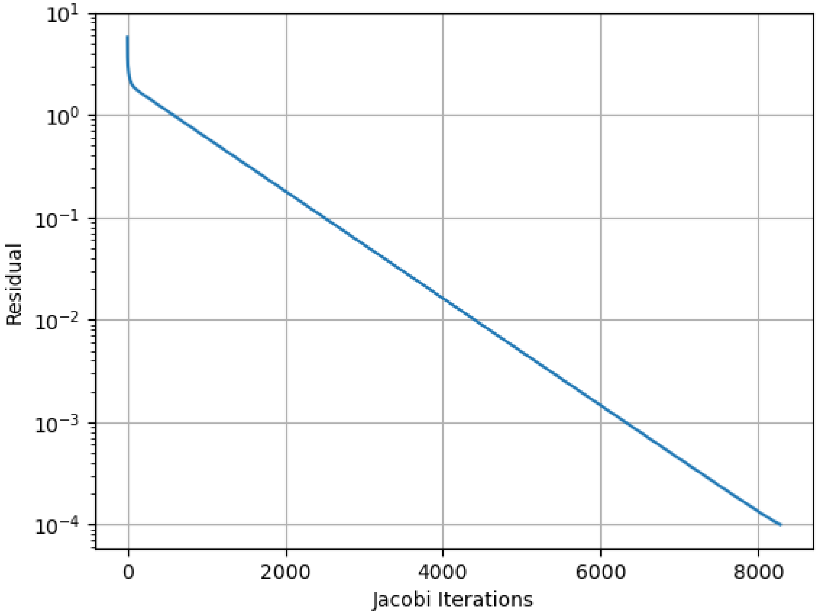


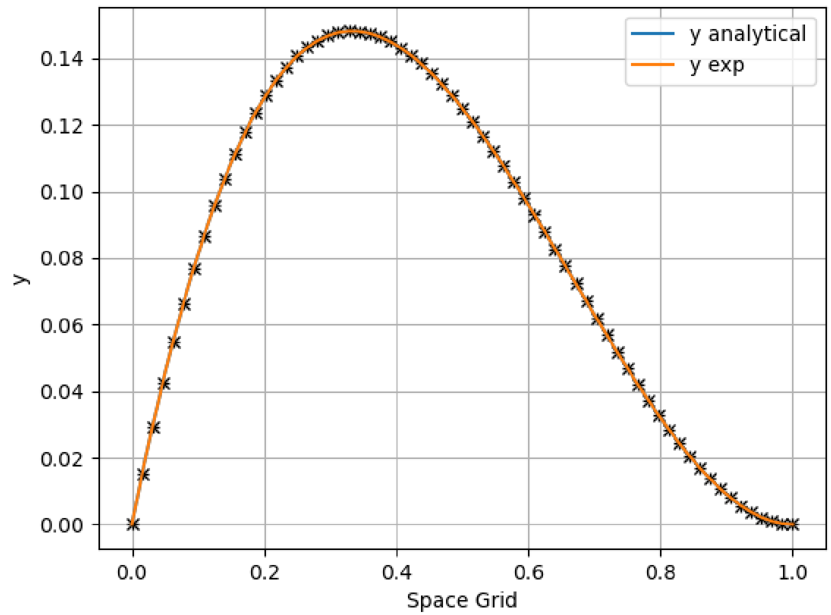
parameter v=0.1
parameter x=0.0625
p tilda
p tilda vector
p final
p analytical
p
Time step
0.00
−0.02
−0.04
−0.06
−0.08
−0.10
−0.12
−0.14
−0.16
0.0
0.2
0.4
0.6
0.8
1.0

SINGLE CYCLE

$$-y'' = f \qquad \text{in } \Omega = (0,1)$$

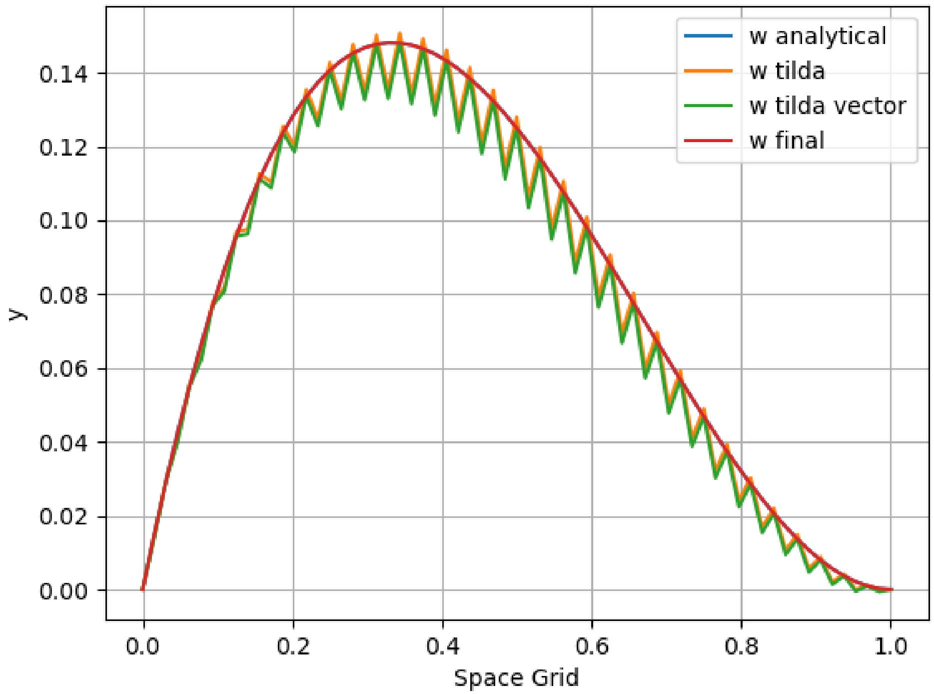
$$y(0) = y(1) = 0$$





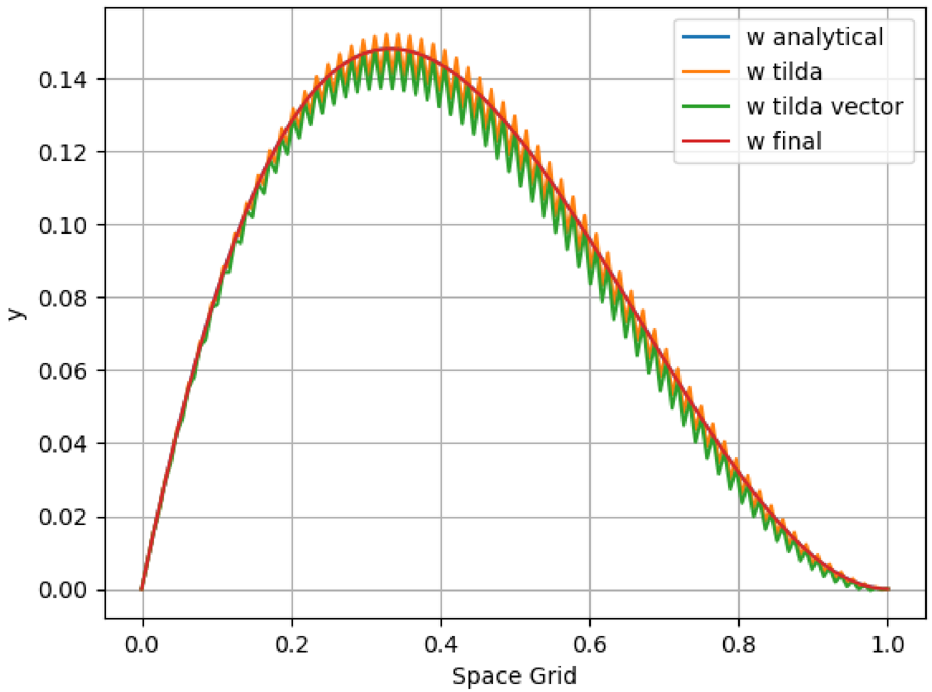
MULTIGRID V-CYCLE

Fine grid: 64
Coarse grid: 32



MULTIGRID V-CYCLE

Fine grid: 128
Coarse grid: 64



VIELEN DANK FÜR IHRE AUFMERKSAMKEIT