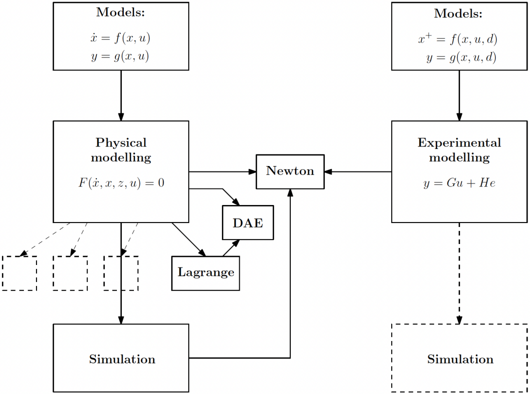
ESS101
Modelling and Simulation, 2025

Lecturer and examiner: Yasemin Bekiroğlu
Course assistant: Ahmet Tekden

Systems & Control Division
Department of Electrical Engineering
Chalmers University of Technology

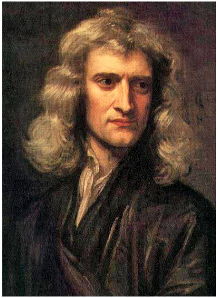
September, 2025

Lecture 8 – The Newton method



Lecture 8 – The Newton method

- Newton method – basic algorithm
- Convergence properties
- Full vs reduced Newton steps
- Quasi-Newton methods
- Newton for optimization



Learning objectives:

- Analyze advanced forms of differential equations used in modelling.
- Understand and implement some of the numerical methods used in simulations.

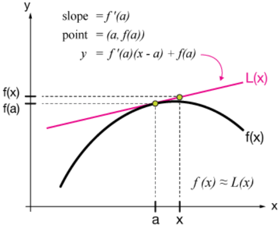
Newton method

The Newton method aims at solving a set of (nonlinear) equations, that we can write as: $\varphi(\mathbf{x}, \mathbf{y}) = 0$, $\varphi \; : \; \mathbb{R}^{n_\mathbf{x}} \; \times \; \mathbb{R}^{n_\mathbf{y}} \; \mapsto \; \mathbb{R}^{n_\mathbf{x}}$

- When $\varphi(\mathbf{x}, \mathbf{y}) = 0$ is nonlinear, finding $\mathbf{x}$ cannot be done explicitly (cannot provide explicit expressions describing $\mathbf{x}$ as a function of $\mathbf{y}$).
- Using the Newton method we compute $\mathbf{x}$ as a function of $\mathbf{y}$ numerically.

Newton method

We make use of linear approximations, first-order Taylor approximation of $\varphi(x, y)$:



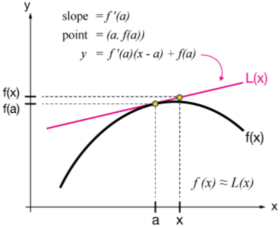
We solve:

$$\boldsymbol{\varphi}\left(\mathbf{x}_{+}, \mathbf{y}\right) \approx \boldsymbol{\varphi}\left(\mathbf{x}, \mathbf{y}\right) + \frac{\partial \boldsymbol{\varphi}\left(\mathbf{x}, \mathbf{y}\right)}{\partial \mathbf{x}}\left(\mathbf{x}_{+} - \mathbf{x}\right) = 0 \tag{1}$$

for $\mathbf{x}_{+}$, based on a given guess $\mathbf{x}$.

Newton method

We make use of linear approximations, first-order Taylor approximation of $\varphi(x, y)$:



We solve:

$$\varphi\left(\mathbf{x}_{+}, \mathbf{y}\right) \approx \varphi\left(\mathbf{x}, \mathbf{y}\right) + \frac{\partial \varphi\left(\mathbf{x}, \mathbf{y}\right)}{\partial \mathbf{x}}\left(\mathbf{x}_{+} - \mathbf{x}\right) = 0 \tag{2}$$

for $\mathbf{x}_{+}$, based on a given guess $\mathbf{x}$.

$$\varphi\left(\mathbf{x}_{+}, \mathbf{y}\right) = \varphi\left(\mathbf{x}, \mathbf{y}\right) + \frac{\partial \varphi\left(\mathbf{x}, \mathbf{y}\right)}{\partial \mathbf{x}}\left(\mathbf{x}_{+} - \mathbf{x}\right) + \mathcal{O}\left(\left\|\mathbf{x}_{+} - \mathbf{x}\right\|^{2}\right) \tag{3}$$

Newton method

We make use of linear approximations, first-order Taylor approximation of $\varphi(x, y)$:
We solve for $\mathbf{x}_+$, based on a given guess $\mathbf{x}$:

$$\boldsymbol{\varphi}\left(\mathbf{x}_+, \mathbf{y}\right) \approx \boldsymbol{\varphi}\left(\mathbf{x}, \mathbf{y}\right) + \frac{\partial \boldsymbol{\varphi}\left(\mathbf{x}, \mathbf{y}\right)}{\partial \mathbf{x}}\left(\mathbf{x}_+ - \mathbf{x}\right) = 0 \tag{4}$$

$\mathbf{x}_+$ can be obtained explicitly:

$$\mathbf{x}_+ = \mathbf{x} - \frac{\partial \boldsymbol{\varphi}\left(\mathbf{x}, \mathbf{y}\right)}{\partial \mathbf{x}}^{-1} \boldsymbol{\varphi}\left(\mathbf{x}, \mathbf{y}\right) \tag{5}$$

Newton method

We make use of linear approximations, first-order Taylor approximation of $\varphi(x, y)$:
We solve for $\mathbf{x}_+$, based on a given guess $\mathbf{x}$:

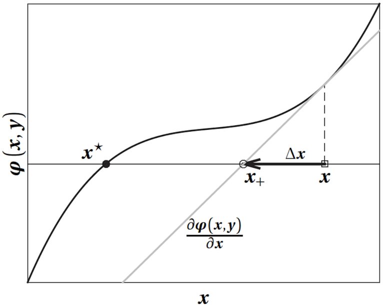
$$\boldsymbol{\varphi}\left(\mathbf{x}_+, \mathbf{y}\right) \approx \boldsymbol{\varphi}\left(\mathbf{x}, \mathbf{y}\right) + \frac{\partial \boldsymbol{\varphi}\left(\mathbf{x}, \mathbf{y}\right)}{\partial \mathbf{x}} \left(\mathbf{x}_+ - \mathbf{x}\right) = 0 \tag{6}$$

$\mathbf{x}_+$ can be obtained explicitly:

$$\mathbf{x}_+ = \mathbf{x} - \frac{\partial \boldsymbol{\varphi}\left(\mathbf{x}, \mathbf{y}\right)}{\partial \mathbf{x}}^{-1} \boldsymbol{\varphi}\left(\mathbf{x}, \mathbf{y}\right) \tag{7}$$

$$\Delta \mathbf{x} = -\frac{\partial \boldsymbol{\varphi}\left(\mathbf{x}, \mathbf{y}\right)}{\partial \mathbf{x}}^{-1} \boldsymbol{\varphi}\left(\mathbf{x}, \mathbf{y}\right) \tag{8}$$

Newton method



Newton method

Algorithm: Full-step Newton method

Input: Variable $\boldsymbol{y}$, initial guess $\boldsymbol{x}$, and tolerance tol

while $\|\boldsymbol{\varphi}(\boldsymbol{x},\boldsymbol{y})\|_\infty \geq \text{tol}$ **do**

Compute

$$\boldsymbol{\varphi}(\boldsymbol{x},\boldsymbol{y}) \quad \text{and} \quad \frac{\partial \boldsymbol{\varphi}(\boldsymbol{x},\boldsymbol{y})}{\partial \boldsymbol{x}} \tag{4.7}$$

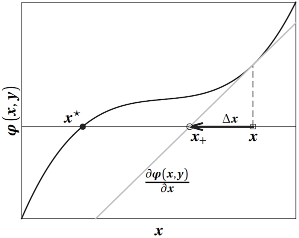
Compute the Newton step

$$\frac{\partial \boldsymbol{\varphi}(\boldsymbol{x},\boldsymbol{y})}{\partial \boldsymbol{x}} \Delta\boldsymbol{x} + \boldsymbol{\varphi}(\boldsymbol{x},\boldsymbol{y}) = 0 \tag{4.8}$$

Take the Newton step

$$\boldsymbol{x} \leftarrow \boldsymbol{x} + \Delta\boldsymbol{x} \tag{4.9}$$

return $\boldsymbol{x}$



Newton method - remarks

- If it converges, the Newton method converges to $\mathbf{x}^\star$ a solution of $\varphi(\mathbf{x}, \mathbf{y}) = 0$

Newton method - remarks

- If it converges, the Newton method converges to $\mathbf{x}^\star$ a solution of $\boldsymbol{\varphi}(\mathbf{x}, \mathbf{y}) = 0$
- Each step of the Newton method requires evaluating the function $\boldsymbol{\varphi}(\mathbf{x}, \mathbf{y})$ and its Jacobian $\frac{\partial \boldsymbol{\varphi}(\mathbf{x})}{\partial \mathbf{x}}$, and solving the linear system (4.8-Newton step).

Newton method - remarks

- If it converges, the Newton method converges to $\mathbf{x}^\star$ a solution of $\boldsymbol{\varphi}(\mathbf{x}, \mathbf{y}) = 0$
- Each step of the Newton method requires evaluating the function $\boldsymbol{\varphi}(\mathbf{x}, \mathbf{y})$ and its Jacobian $\frac{\partial \boldsymbol{\varphi}(\mathbf{x})}{\partial \mathbf{x}}$, and solving the linear system (4.8-Newton step).
- The Newton method requires that the square Jacobian matrix $\frac{\partial \boldsymbol{\varphi}(\mathbf{x},\mathbf{y})}{\partial \mathbf{x}}$ is full rank (i.e. invertible), in order for the linear system (4.8-Newton step) to be well-posed.

Newton method - remarks

- If it converges, the Newton method converges to $\mathbf{x}^\star$ a solution of $\boldsymbol{\varphi}(\mathbf{x}, \mathbf{y}) = 0$
- Each step of the Newton method requires evaluating the function $\boldsymbol{\varphi}(\mathbf{x}, \mathbf{y})$ and its Jacobian $\frac{\partial \boldsymbol{\varphi}(\mathbf{x})}{\partial \mathbf{x}}$, and solving the linear system (4.8-Newton step).
- The Newton method requires that the square Jacobian matrix $\frac{\partial \boldsymbol{\varphi}(\mathbf{x},\mathbf{y})}{\partial \mathbf{x}}$ is full rank (i.e. invertible), in order for the linear system (4.8-Newton step) to be well-posed.
- If function $\boldsymbol{\varphi}(\mathbf{x}, \mathbf{y})$ is linear in $\mathbf{x}$ (and well posed), then the Newton method finds the solution $\mathbf{x}^\star$ in one step. It is then fully equivalent to solving the linear system.

Convergence Rate

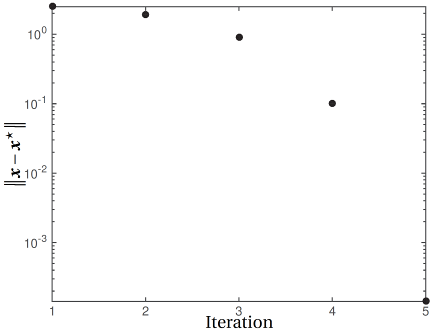
Assume that the Newton iteration converges, its convergence rate, i.e how quickly $\|\mathbf{x} - \mathbf{x}^\star\|$ decreases is addressed as:

Theorem

if the full step Newton iteration converges, then it converges at a quadratic rate, i.e.

$$\|\mathbf{x}_+ - \mathbf{x}^\star\| \leq C \cdot \|\mathbf{x} - \mathbf{x}^\star\|^2 \tag{9}$$

for some constant $C > 0$.

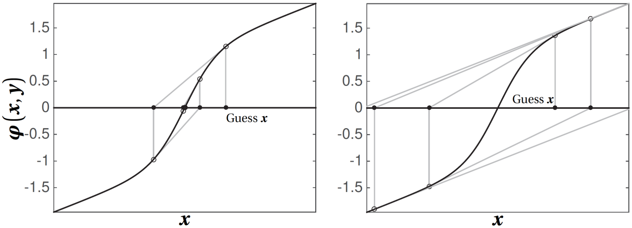


Convergence Rate

x_n	x	$f(x) = x^2 - 2$	$f'(x) = 2x$	$x - \frac{f(x)}{f'(x)}$	1.4142135623731
x_1	1	-1	2	$1 - \frac{-1}{2} = 3/2$	$\underline{1}.5000000000000$
x_2	$\frac{3}{2}$	$\frac{1}{4}$	3	$\frac{3}{2} - \frac{1/4}{3} = \frac{17}{2}$	$\underline{1.41}66666666667$
x_3	$\frac{17}{12}$	$\frac{1}{144}$	$\frac{17}{6}$	$\frac{17}{12} - \frac{1/144}{17/6} = \frac{577}{408}$	$\underline{1.414215}6862745$
x_4	$\frac{577}{408}$	$\frac{1}{166464}$	$\frac{577}{204}$	$\frac{665857}{470832}$	$\underline{1.41421356237}47$

Convergence

In order for the full-step Newton iteration to converge, it should be provided with an initial guess that is **close enough** to a solution $\mathbf{x}^\star$.



Reduced Newton Step

- The full-step Newton algorithm can diverge if the initial guess provided to the Newton iteration is too far from a solution x^*.
- A careful selection of reduced Newton steps, i.e. modifications of x in the direction of the Newton step Δx, scaled down, must converge, as long as the Newton steps Δx exist.

Algorithm: Newton method with reduced steps

Input: Variable $\boldsymbol{y}$, initial guess $\boldsymbol{x}$, and tolerance tol

while $\|\boldsymbol{\varphi}(\boldsymbol{x},\boldsymbol{y})\|_\infty \geq \text{tol}$ **do**

Compute

$$\boldsymbol{\varphi}(\boldsymbol{x},\boldsymbol{y}) \quad \text{and} \quad \frac{\partial \boldsymbol{\varphi}(\boldsymbol{x},\boldsymbol{y})}{\partial \boldsymbol{x}}$$

Compute the Newton step

$$\frac{\partial \boldsymbol{\varphi}(\boldsymbol{x},\boldsymbol{y})}{\partial \boldsymbol{x}} \Delta \boldsymbol{x} + \boldsymbol{\varphi}(\boldsymbol{x},\boldsymbol{y}) = 0$$

Select step size $t \in]0,1]$

Take Newton step

$$\boldsymbol{x} \leftarrow \boldsymbol{x} + t\Delta \boldsymbol{x}$$

return $\boldsymbol{x} \approx \boldsymbol{x}^\star$

Reduced Newton Step

- Taking reduced steps is guaranteed to converge to a solution x^*, as long as the Newton steps Δx exist.

Reduced Newton Step

- Taking reduced steps is guaranteed to converge to a solution x^*, as long as the Newton steps Δx exist.
- When reduced steps (i.e. using $t < 1$) are necessary, then the quadratic convergence rate detailed is lost.

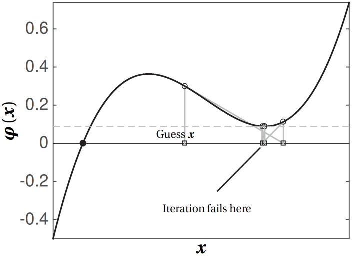
Reduced Newton Step

- Taking reduced steps is guaranteed to converge to a solution x^*, as long as the Newton steps Δx exist.
- When reduced steps (i.e. using t < 1) are necessary, then the quadratic convergence rate detailed is lost.
- When x is far from a solution x^*, then reduced steps ($t < 1$) are necessary, and the algorithm converges slowly. The resulting convergence rate can be very poor, even though it is often close to linear.

Reduced Newton Step

- Taking reduced steps is guaranteed to converge to a solution x^*, as long as the Newton steps Δx exist.
- When reduced steps (i.e. using t < 1) are necessary, then the quadratic convergence rate detailed is lost.
- When x is far from a solution x^*, then reduced steps ($t < 1$) are necessary, and the algorithm converges slowly. The resulting convergence rate can be very poor, even though it is often close to linear.
- After iterations, x becomes close enough to x^* and full steps ($t = 1$) are acceptable. The convergence then becomes quadratic.

Reduced Newton Step



Newton iteration with reduced steps on a nonlinear, scalar function $\varphi(x)$ (five steps are displayed here). Here the iteration does not diverge, but is fails at a point where $\frac{\partial \varphi(x,y)}{\partial x} = 0$. At this point, the linear system (4.8) does not have a well-defined solution and the Newton step Δx ceases to exist.

- If the Newton iteration was started closer to the solution x^* (black dot in the graph), then it would converge.

Jacobian Approximation

In some applications of the Newton iteration, the evaluation of the Jacobian $\frac{\partial \boldsymbol{\varphi}(\boldsymbol{x})}{\partial \boldsymbol{x}}$ is very expensive. It can then be useful to consider using an approximation that is less expensive to evaluate. Let us label this approximation:

$$M \approx \frac{\partial \boldsymbol{\varphi}(\boldsymbol{x})}{\partial \boldsymbol{x}} \tag{4.43}$$

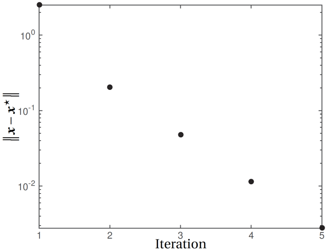
The resulting *Newton-type* step reads as:

$$\Delta \boldsymbol{x} = -M^{-1} \boldsymbol{\varphi}(\boldsymbol{x}) \tag{4.44}$$

Theorem 7. *the convergence of the full-step Newton method with an approximate Jacobian follows:*

$$\left\| \boldsymbol{x}_+ - \boldsymbol{x}^\star \right\| \leq \left(\kappa + \frac{c}{2} \left\| \boldsymbol{x} - \boldsymbol{x}^\star \right\| \right) \left\| \boldsymbol{x} - \boldsymbol{x}^\star \right\| \tag{4.45}$$

for some constants $c, \kappa > 0$.



Newton for Optimization

$$\min_{\boldsymbol{x}} \; \Phi\left(\boldsymbol{x}, \boldsymbol{y}\right)$$

$$\nabla_{\boldsymbol{x}} \Phi\left(\boldsymbol{x}, \boldsymbol{y}\right) = 0$$

Algorithm: Newton method with reduced steps

Input: Variable $\boldsymbol{y}$, initial guess $\boldsymbol{x}$, and tolerance tol

while $\|\boldsymbol{\varphi}\left(\boldsymbol{x}, \boldsymbol{y}\right)\|_\infty \geq$ tol **do**

Compute

$$\boldsymbol{\varphi}\left(\boldsymbol{x}, \boldsymbol{y}\right) \quad \text{and} \quad \frac{\partial \boldsymbol{\varphi}\left(\boldsymbol{x}, \boldsymbol{y}\right)}{\partial \boldsymbol{x}}$$

Compute the Newton step

$$\frac{\partial \boldsymbol{\varphi}\left(\boldsymbol{x}, \boldsymbol{y}\right)}{\partial \boldsymbol{x}} \Delta \boldsymbol{x} + \boldsymbol{\varphi}\left(\boldsymbol{x}, \boldsymbol{y}\right) = 0$$

Select step size $t \in]0, 1]$

Take Newton step

$$\boldsymbol{x} \leftarrow \boldsymbol{x} + t\Delta \boldsymbol{x}$$

return $\boldsymbol{x} \approx \boldsymbol{x}^\star$

Algorithm: Newton method for optimization

Input: Variable $\boldsymbol{y}$, initial guess $\boldsymbol{x}$, and tolerance tol

while $\|\nabla_{\boldsymbol{x}} \Phi\left(\boldsymbol{x}, \boldsymbol{y}\right)\|_\infty \geq$ tol **do**

Compute

$$\nabla_{\boldsymbol{x}} \Phi\left(\boldsymbol{x}, \boldsymbol{y}\right) \quad \text{and} \quad \nabla^2_{\boldsymbol{x}} \Phi\left(\boldsymbol{x}, \boldsymbol{y}\right)$$

Compute the Newton step

$$\nabla^2_{\boldsymbol{x}} \Phi\left(\boldsymbol{x}, \boldsymbol{y}\right) \Delta \boldsymbol{x} + \nabla_{\boldsymbol{x}} \Phi\left(\boldsymbol{x}, \boldsymbol{y}\right) = 0$$

Select step size $t \in]0, 1]$

Take Newton step

$$\boldsymbol{x} \leftarrow \boldsymbol{x} + t\Delta \boldsymbol{x}$$

return $\boldsymbol{x} \approx \boldsymbol{x}^\star$

The Newton method – summary

The *Newton method* aims at solving the equation $\varphi(x, y) = 0$ by iteratively performing the following computations:

1. Compute $\varphi(x, y)$ and $\dfrac{\partial \varphi(x, y)}{\partial x}$
2. Compute the *Newton step* Δx by solving $\dfrac{\partial \varphi(x, y)}{\partial x} \Delta x + \varphi(x, y) = 0$
3. Update x: $x \leftarrow x + t \cdot \Delta x$ *full step:* $t = 1$; *reduced step:* $0 < t < 1$

A *quasi-Newton method* uses an approximation of the Jacobian: $M \approx \frac{\partial \varphi(x,y)}{\partial x}$.

Properties:

- Newton with full steps: quadratic convergence close to the solution; with reduced steps: slow convergence.
- Quasi-Newton with full steps: linear convergence close to the solution if M good enough; with reduced steps: slow convergence.
- The Newton iteration fails if the Jacobian $\frac{\partial \varphi(x,y)}{\partial x}$ becomes singular.

Newton method for optimization

Consider the unconstrained optimization problem:

$$\min_x \Phi(x,y); \qquad \Phi : \; \mathrm{R}^{n_x} \times \mathrm{R}^{n_y} \to \mathrm{R}$$

Candidate solutions (local extrema) must fulfil the *necessary* condition

$$\nabla_x \Phi(x,y) = 0.$$

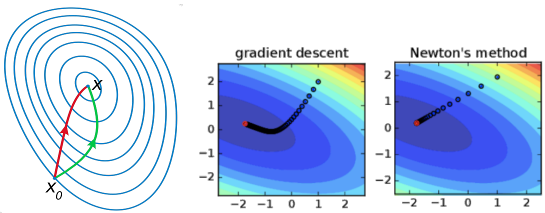
The condition is *sufficient* if $\Phi(x,y)$ is *convex*, i.e. if the *Hessian* $\nabla_x^2 \Phi(x,y)$ is positive definite.

Applying Newton to the necessary condition (using $\varphi(x,y) = \nabla_x \Phi(x,y)$) gives the (full) Newton step Δx from the equation

$$\nabla_x^2 \Phi(x,y) \cdot \Delta x + \nabla_x \Phi(x,y) = 0$$

A quasi-Newton method is obtained by using any positive definite approximation M of the Hessian ($M = I$ gives a *gradient* or *steepest-descent* direction).

Newton takes a more direct route in optimization



A comparison of gradient descent (green) and Newton's method (red) for minimizing a function (with small step sizes). Newton's method uses curvature information (i.e. the second derivative) to take a more direct route.

Example: nonlinear least-squares

Consider the following optimization problem, arising in e.g. system identification:

$$\min_x \Phi(x, y), \qquad \Phi(x, y) = \frac{1}{2}\|\phi(x) - y\|^2.$$

To apply the Newton method, the gradient and Hessian of Φ is needed.

Using the *Gauss-Newton* approximation of the Hessian, we have:

$$\begin{aligned}
\nabla_x \Phi(x, y) &= \nabla_x \phi(x)(\phi(x) - y) \\
\nabla_{\mathbf{x}}^2 \Phi(\mathbf{x}, \mathbf{y}) &= \nabla_{\mathbf{x}} \phi(\mathbf{x}) \nabla_{\mathbf{x}} \phi(\mathbf{x})^\top + \left[\nabla_{\mathbf{x}_i, \mathbf{x}_j} \phi(\mathbf{x}) (\phi(\mathbf{x}) - \mathbf{y})\right]_{i,j} \\
\nabla_x^2 \Phi(x, y) &\approx \nabla_x \phi(x) \nabla_x \phi(x)^\top
\end{aligned}$$

- The approximation is valid if $\phi(\mathbf{x})$ is not very nonlinear, such that its second-order derivatives $\nabla_{\mathbf{x}_i, \mathbf{x}_j} \phi(\mathbf{x})$ are small.
- Often a good approximation when $\phi(x) - y$ is small.
- Automatically gives a positive semi-definite matrix.
- *Regularization* is used to secure a pos. def. approximation of the Hessian:

$$M = \nabla_x \phi(x) \nabla_x \phi(x)^\top + \alpha \cdot I, \quad \alpha > 0$$

Newton method for system identification

Example (PEM with quadratic cost)

$$\hat{\theta}_N = \arg\min_{\theta} V_N(\theta) = \arg\min_{\theta} \frac{1}{N} \sum_{t=1}^{N} \big(y(t) - \hat{y}(t|t-1,\theta)\big)^2$$

Using the Gauss-Newton approximation, we obtain

$$\nabla_\theta V_N(\theta) = -\frac{2}{N} \sum_{t=1}^{N} \nabla_\theta \hat{y}(t|t-1,\theta)\big(y(t) - \hat{y}(t|t-1,\theta)\big)$$

$$\nabla_\theta^2 V_N(\theta) \approx \frac{2}{N} \sum_{t=1}^{N} \nabla_\theta \hat{y}(t|t-1,\theta)\, \nabla_\theta \hat{y}(t|t-1,\theta)^T$$

The (full) Newton step is obtained by solving

$$\Big[\frac{1}{N} \sum_{t=1}^{N} \nabla_\theta \hat{y}(t|t-1,\theta)\, \nabla_\theta \hat{y}(t|t-1,\theta)^T\Big] \Delta\theta = \frac{1}{N} \sum_{t=1}^{N} \nabla_\theta \hat{y}(t|t-1,\theta)\big(y(t) - \hat{y}(t|t-1,\theta)\big)$$

Compare with the normal equations (which give $\hat{\theta}_N$ in one step)!

Summary:

- Exact reduced Newton steps $\Delta\mathbf{x}$ improves φ for sufficiently small step sizes $t \in]0, 1]$
  - Inexact reduced Newton steps $\Delta\mathbf{x}$ improve φ for a sufficiently small step size $t \in]0, 1]$ <u>if</u> M is sufficiently close to $\frac{\partial\varphi}{\partial\mathbf{x}}$. In the context of optimization, $M > 0$ and sufficiently small steps $t \in]0, 1]$ reduce the cost function Φ.
- Exact full ($t = 1$) Newton steps converge quadratically if close enough to the solution
  - Inexact full ($t = 1$) Newton steps converge linearly if close enough to the solution <u>and</u> if the Jacobian approximation is sufficiently good.
- The Newton iteration fails if $\frac{\partial\varphi}{\partial\mathbf{x}}$ becomes singular
- Newton methods with reduced steps converge in two phases: damped (slow) phase when reduced steps ($t < 1$) are needed, quadratic/ linear when full steps are possible.