# Gradient-based Methods for Optimization. Part II.

**Nathan L. Gibson**

gibsonn@math.oregonstate.edu

**Department of Mathematics**
**Oregon State University**

# Summary from Last Time

- Unconstrained Optimization
  - Nonlinear Least Squares
    - Parameter ID Problem

Sample Problem:

$$u'' + cu' + ku = 0; u(0) = u_0; u'(0) = 0 \qquad (1)$$

Assume data $\{u_j\}_{j=0}^{M}$ is given for some times $t_j$ on the interval $[0, T]$. Find $x$=$[c, k]^T$ such that the following objective function is minimized:

$$f(x) = \frac{1}{2}\sum_{j=1}^{M} |u(t_j; x) - u_j|^2 .$$

# Summary Continued

Update step

$$x_{k+1} = x_k + s_k$$

- Newton's Method – quadratic model
  - Gauss-Newton – neglect 2nd order terms
- Steepest Descent – always descent direction
- Levenberg-Marquardt – like a weighted average of GN and SD with parameter $\nu$

# Summary of Methods

- Newton:

$$m_k^N(x) = f(x_k) + \nabla f(x_k)^T(x - x_k) + \frac{1}{2}(x - x_k)^T \nabla^2 f(x_k)(x - x_k)$$

- Gauss-Newton:

$$m_k^{GN}(x) = f(x_k) + \nabla f(x_k)^T(x - x_k) + \frac{1}{2}(x - x_k)^T R'(x_k)^T R'(x_k)(x - x_k)$$

- Steepest Descent:

$$m_k^{SD}(x) = f(x_k) + \nabla f(x_k)^T(x - x_k) + \frac{1}{2}(x - x_k)^T \frac{1}{\lambda_k} I(x - x_k)$$

- Levenberg-Marquardt:

$$m_k^{LM}(x) = f(x_k) + \nabla f(x_k)^T(x - x_k) + \frac{1}{2}(x - x_k)^T \left(R'(x_k)^T R'(x_k) + \nu_k I\right)(x - x_k)$$

$$0 = \nabla m_k(x) \implies H_k s_k = -\nabla f(x_k)$$

# Levenberg-Marquardt Idea

- If iterate is not close enough to minimizer so that GN does not give a descent direction, increase $\nu$ to take more of a SD direction.
- As you get closer to minimizer, decrease $\nu$ to take more of a GN step.
  - For zero-residual problems, GN converges quadratically (if at all)
  - SD converges linearly (guaranteed)

# LM Alternative Perspective

- Approximate Hessian may not be positive definite (or well-conditioned), increase $\nu$ to add regularity.
- As you get closer to minimizer, Hessian will become positive definite. Decrease $\nu$ as less regularization is necessary.
- Regularized problem is "nearby problem", want to solve actual problem as soon as feasible.

# Step Length

## Steepest Descent Method

- We define the *steepest descent direction* to be $d_k = -\nabla f(x_k)$. This defines a direction but not a *step length*.
- We define the Steepest Descent update step to be $s_k^{SD} = \lambda_k d_k$ for some $\lambda_k > 0$.
- We would like to choose $\lambda_k$ so that $f(x)$ *decreases sufficiently*.
- Could ask simply that

$$f(x_{k+1}) < f(x_k)$$

# Predicted Reduction

Consider a linear model of $f(x)$

$$m_k(x) = f(x_k) + \nabla f(x_k)^T (x - x_k).$$

Then the *predicted reduction* using the Steepest Descent step $(x_{k+1} = x_k - \lambda_k \nabla f(x_k))$ is

$$pred = m_k(x_k) - m_k(x_{k+1}) = \lambda_k \|\nabla f(x_k)\|^2.$$

The *actual reduction* in $f$ is

$$ared = f(x_k) - f(x_{k+1}).$$

# Sufficient Decrease

We define a sufficient decrease to be when

$$ared > \alpha \, pred,$$

where $\alpha \in (0, 1)$ (e.g., $10^{-4}$ or so).
Note: $\alpha = 0$ is simple decrease.

# Armijo Rule

We can define a strategy for determining the step length in terms of a sufficient decrease criteria as follows:
Let $\lambda = \beta^m$, where $\beta \in (0,1)$ (think $\frac{1}{2}$) and $m \geq 0$ is the smallest integer such that

$$ared > \alpha\, pred,$$

where $\alpha \in (0,1)$.

# Line Search

- The *Armijo Rule* is an example of a line search: Search on a ray from $x_k$ in direction of locally decreasing $f$.
- Armijo procedure is to start with $m = 0$ then increment $m$ until sufficient decrease is achieved, i.e., $\lambda = \beta^m = 1, \beta, \beta^2, \ldots$
- This approach is also called “backtracking” or performing “pullbacks”.
- For each $m$ a new function evaluation is required.

# Damped Gauss-Newton

- Armijo Rule applied to the Gauss-Newton step is called the *Damped Gauss-Newton Method*.
- Recall

$$d^{GN} = -\left(R'(x)^T R'(x)\right)^{-1} R'(x)^T R(x).$$

- Note that if $R'(x)$ has full column rank, then

$$0 > \nabla f(x)^T d^{GN} =$$

$$-\left(R'(x)^T R(x)\right)^T \left(R'(x)^T R'(x)\right)^{-1} R'(x)^T R(x)$$

so the GN direction is a descent direction.

# Damped Gauss-Newton Step

Thus the step for Damped Gauss-Newton is

$$s^{DGN} = \beta^m d^{GN}$$

where $\beta \in (0, 1)$ and $m$ is the smallest non-negative integer to guarantee sufficient decrease.

# Levenberg-Marquardt-Armijo

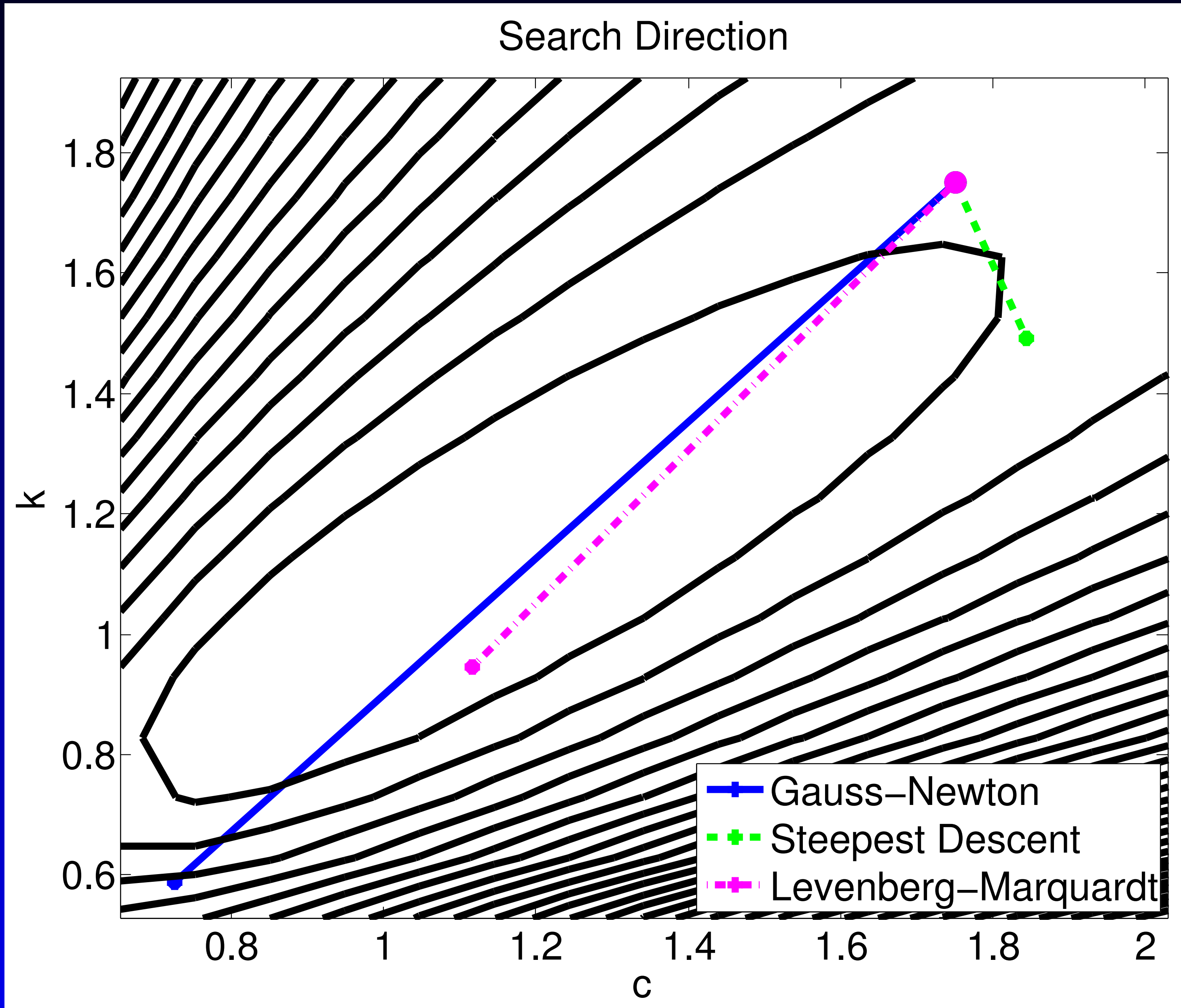
- If $R'(x)$ does not have full column rank, or if the matrix $R'(x)^T R'(x)$ may be ill-conditioned, you should be using Levenberg-Marquardt.
- The LM direction is a descent direction.
- Line search can be applied.
- Can show that if $\nu_k = O(\|R(x_k)\|)$ then LMA converges quadratically for (nice) zero residual problems.

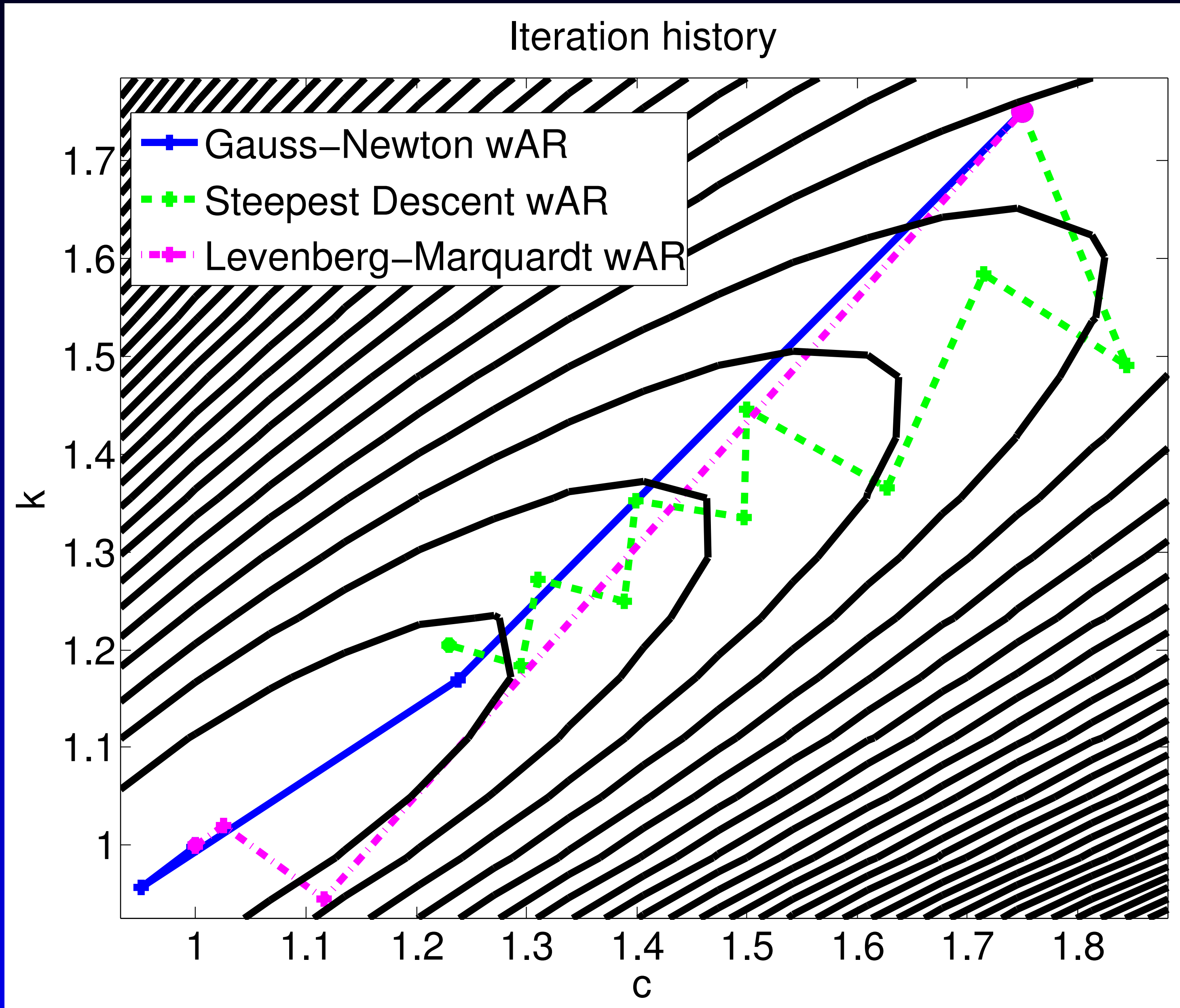
# Numerical Example

- Recall

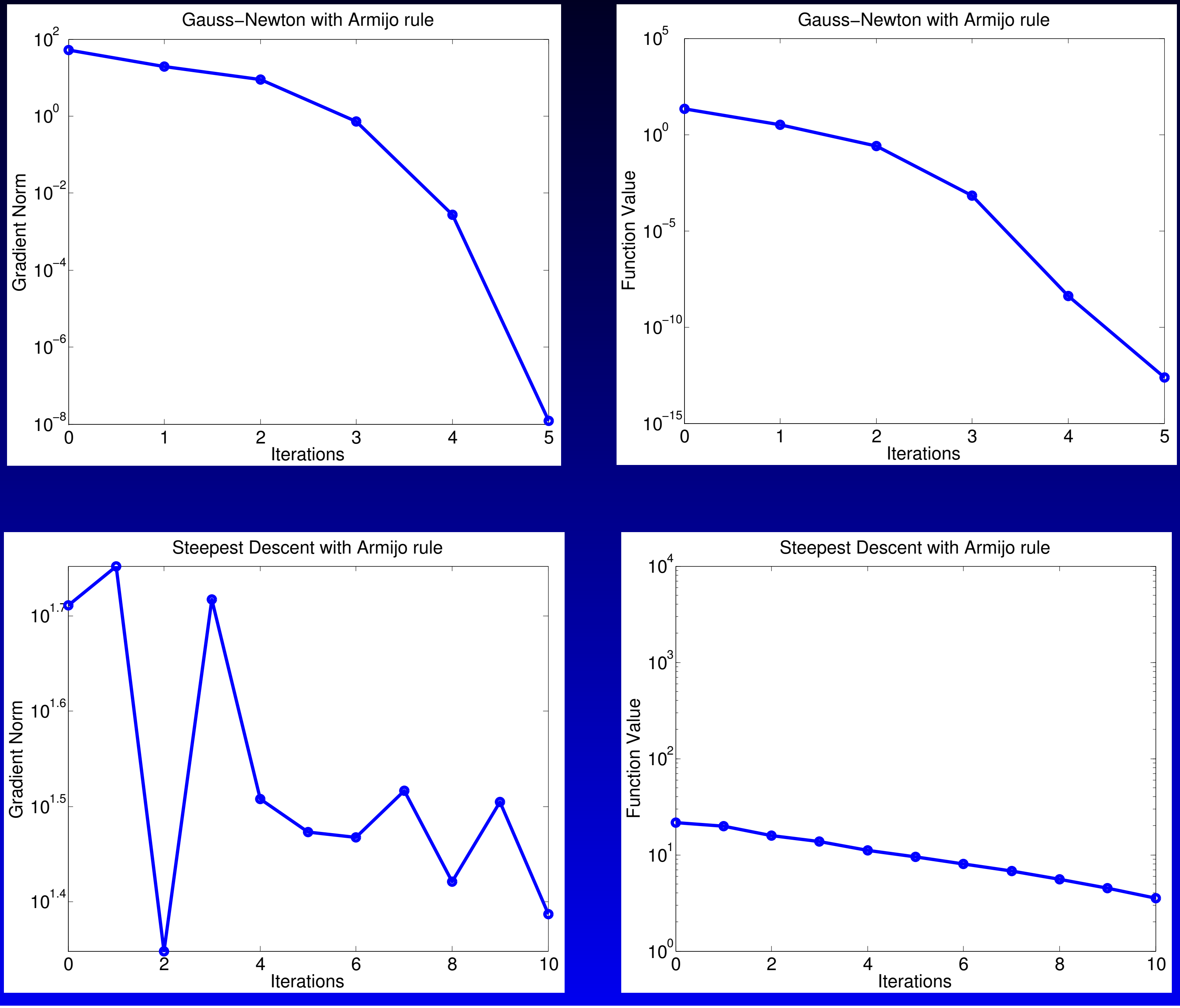
$$u'' + cu' + ku = 0; u(0) = u_0; u'(0) = 0.$$

- Let the true parameters be $x^* = [c, k]^T = [1, 1]^T$. Assume we have $M = 100$ data $u_j$ from equally spaced time points on $[0, 10]$.
- We will use the initial iterate $x_0 = [3, 1]^T$ with Steepest Descent, Gauss-Newton and Levenberg-Marquardt methods using the Armijo Rule.

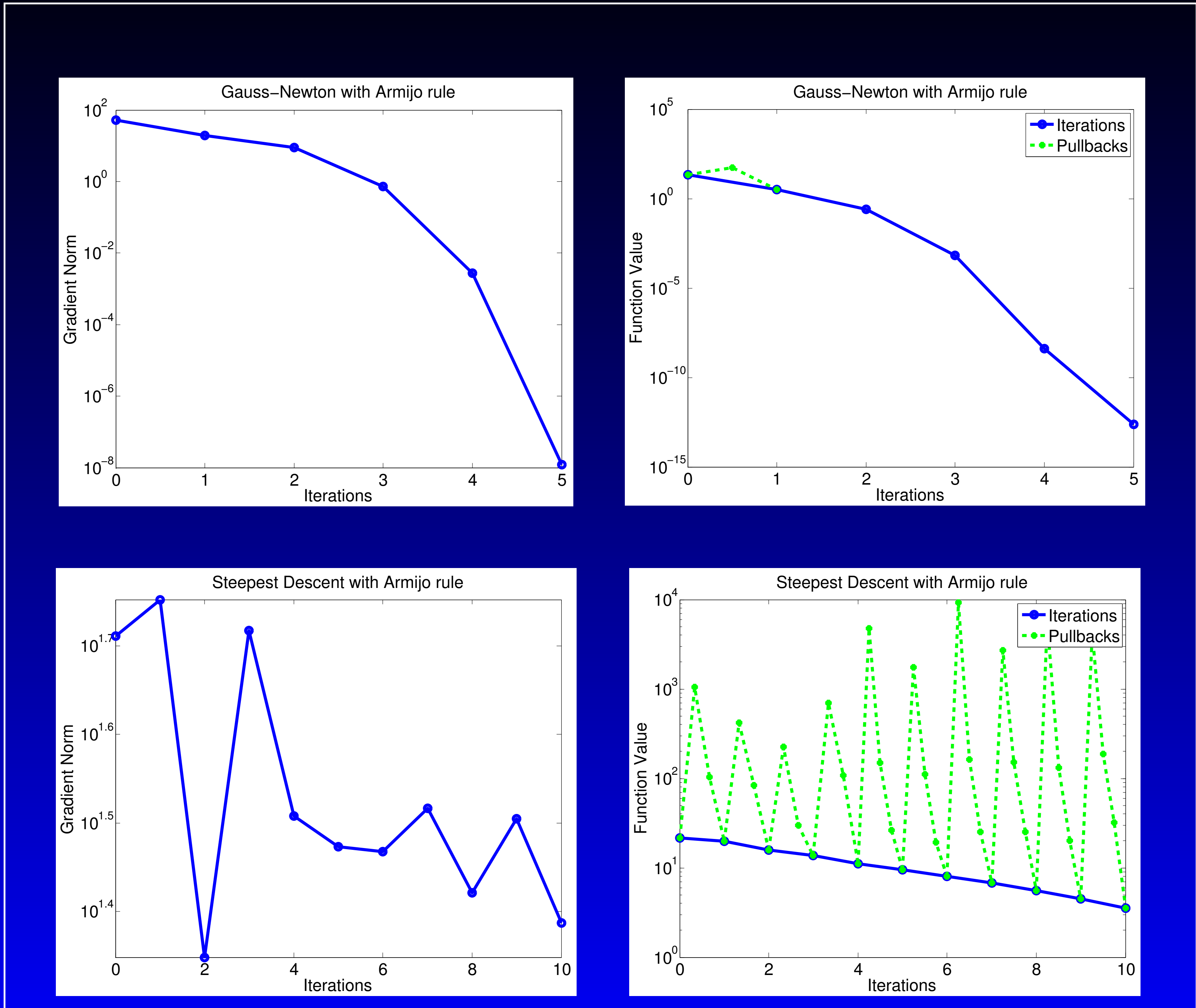

Search Direction
k
c
1.8
1.6
1.4
1.2
1
0.8
0.6
0.8
1
1.2
1.4
1.6
1.8
2
Gauss−Newton
Steepest Descent
Levenberg−Marquardt



Iteration history
Gauss−Newton wAR
Steepest Descent wAR
Levenberg−Marquardt wAR
1.7
1.6
1.5
1.4
1.3
1.2
1.1
1
k
1
1.1
1.2
1.3
1.4
1.5
1.6
1.7
1.8
c

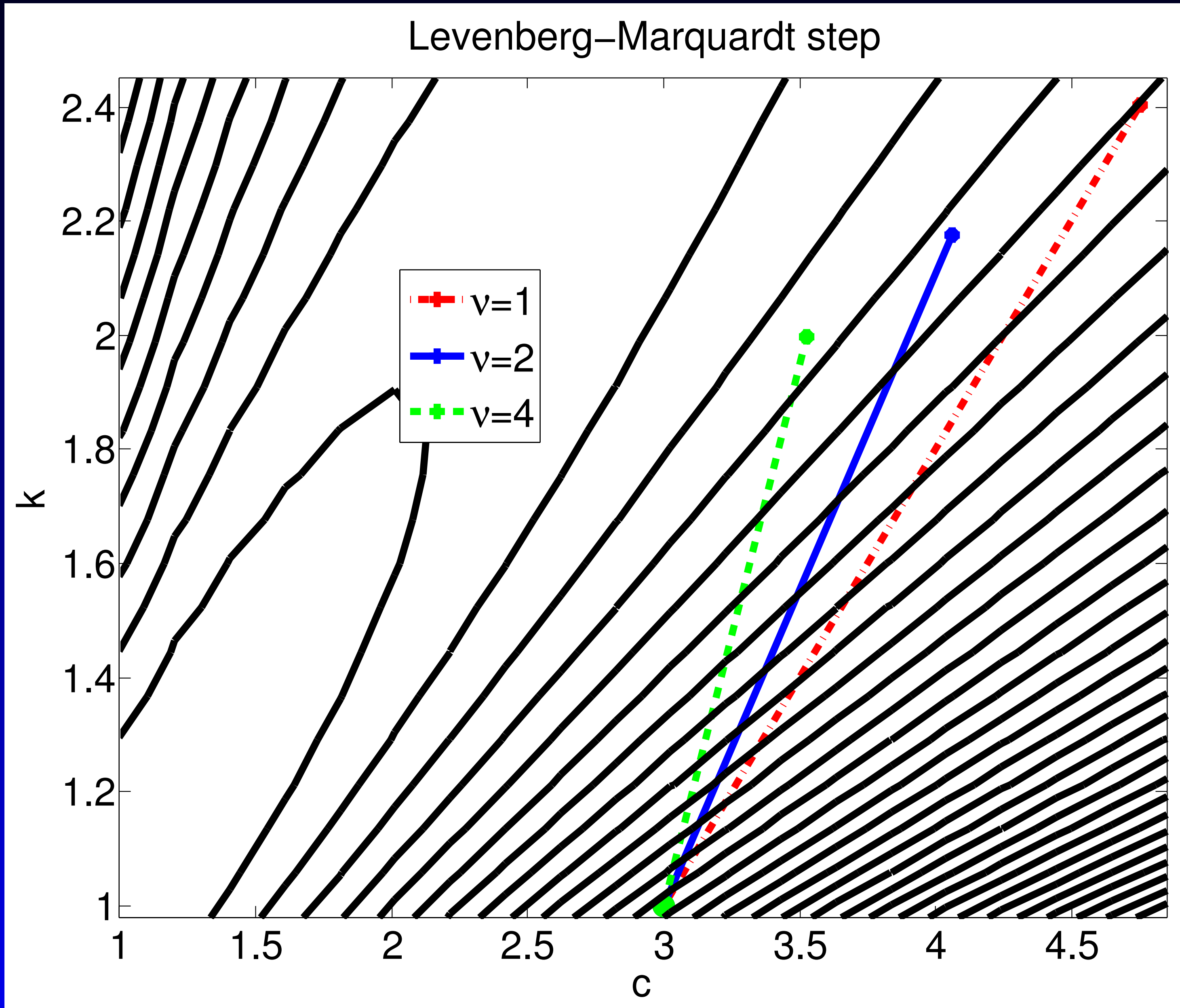


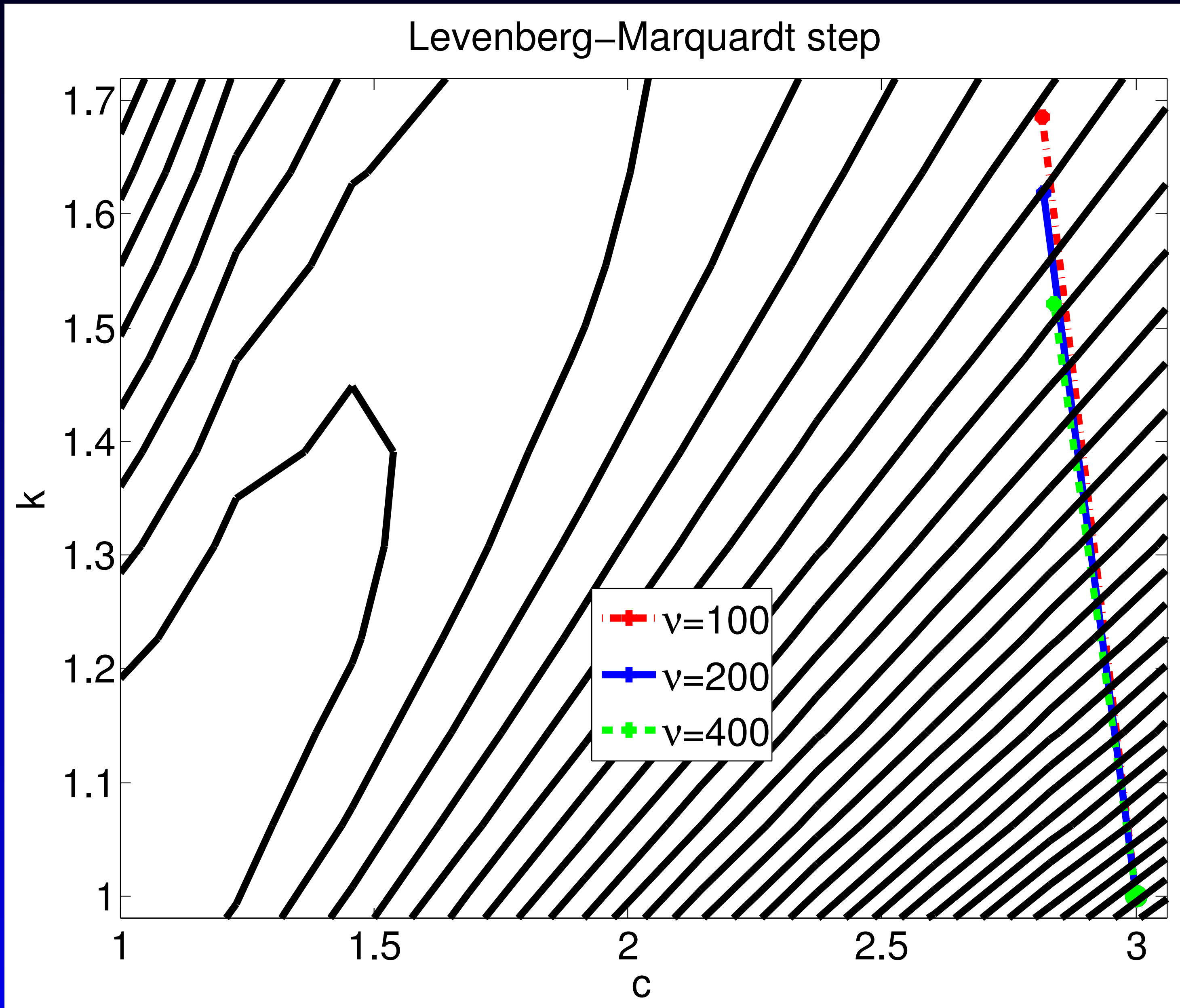
Gauss−Newton with Armijo rule
Gradient Norm
Iterations
Gauss−Newton with Armijo rule
Function Value
Iterations
Steepest Descent with Armijo rule
Gradient Norm
Iterations
Steepest Descent with Armijo rule
Function Value
Iterations


Gauss−Newton with Armijo rule
Gradient Norm
Iterations
Gauss−Newton with Armijo rule
Function Value
Iterations
Pullbacks
Steepest Descent with Armijo rule
Gradient Norm
Iterations
Steepest Descent with Armijo rule
Function Value
Iterations
Pullbacks

# Word of Caution for LM

- Note that blindly increasing $\nu$ until a sufficient decrease criteria is satisfied is NOT a good idea (nor is it a line search).
- Changing $\nu$ changes direction as well as step length.
- Increasing $\nu$ does insure your direction is descending.
- But, increasing $\nu$ too much makes your step length small.


Levenberg−Marquardt step
ν=1
ν=2
ν=4
k
c
2.4
2.2
2
1.8
1.6
1.4
1.2
1
1
1.5
2
2.5
3
3.5
4
4.5



Levenberg−Marquardt step
1.7
1.6
1.5
1.4
1.3
1.2
1.1
1
k
1
1.5
2
2.5
3
c
ν=100
ν=200
ν=400

# Line Search Improvements

**Step length control with polynomial models**

- If $\lambda = 1$ does not give sufficient decrease, use $f(x_k)$, $f(x_k + d)$ and $\nabla f(x_k)$ to build a quadratic model of
$$\xi(\lambda) = f(x_k + \lambda d)$$
- Compute the $\lambda$ which minimizes model of $\xi$.
- If this fails, create cubic model.
- If this fails, switch back to Armijo.
- *Exact line search* is (usually) not worth the cost.

# Trust Region Methods

- Let $\Delta$ be the radius of a ball about $x_k$ inside which the quadratic model

$$m_k(x) = f(x_k) + \nabla f(x_k)^T (x - x_k) + \frac{1}{2}(x - x_k)^T H_k (x - x_k)$$

  can be "trusted" to accurately represent $f(x)$.
- $\Delta$ is called the *trust region radius*.
- $\mathcal{T}(\Delta) = \{x | \, \|x - x_k\| \leq \Delta\}$ is called the *trust region*.

# Trust Region Problem

- We compute a trial solution $x_t$, which may or may not become our next iterate.
- We define the trial solution in terms of a trial step $x_t = x_k + s_t$.
- The trial step is the (approximate) solution to the *trust region problem*

$$\min_{\|s\| \leq \Delta} m_k(x_k + s).$$

I.e., find the trial solution in the trust region which minimizes the quadratic model of $f$.

# Unidirectional TR Algorithm

Suppose we limit our search of $s_t$ to the direction of $d^{SD}$. Then the trust region problem becomes

$$\min_{x_k - \lambda \nabla f(x_k) \in \mathcal{T}(\Delta_k)} m_k(x_k - \lambda \nabla f(x_k)),$$

$$m_k(x_k - \lambda \nabla f(x_k)) = f(x_k) + \nabla f(x_k)^T(-\lambda \nabla f(x_k)) + \frac{1}{2}(-\lambda \nabla f(x_k))^T H_k(-\lambda \nabla f(x_k))$$

$$\hat{\lambda} = \min\left(\frac{\|\nabla f(x_k)\|^2}{\nabla f(x_k)^T H_k \nabla f(x_k)}, \frac{\Delta_c}{\|\nabla f(x_k)\|}\right)$$

# Changing Trust Region

- Test the trial solution $x_t$ using *predicted* and *actual* reductions.
- If $\mu = ared/pred$ too low, reject trial step and decrease trust region radius.
- If $\mu$ sufficiently high, we can accept the trial step, and possibly even increase the trust region radius (becoming more aggressive).

# Exact Solution to TR Problem

**Theorem 1** *Let $g \in \mathbb{R}^N$ and let $A$ be a symmetric $N \times N$ matrix. Let*

$$m(s) = g^T s + s^T As/2.$$

*Then a vector $s$ is a solution to*

$$\min_{\|s\| \leq \Delta} m(s)$$

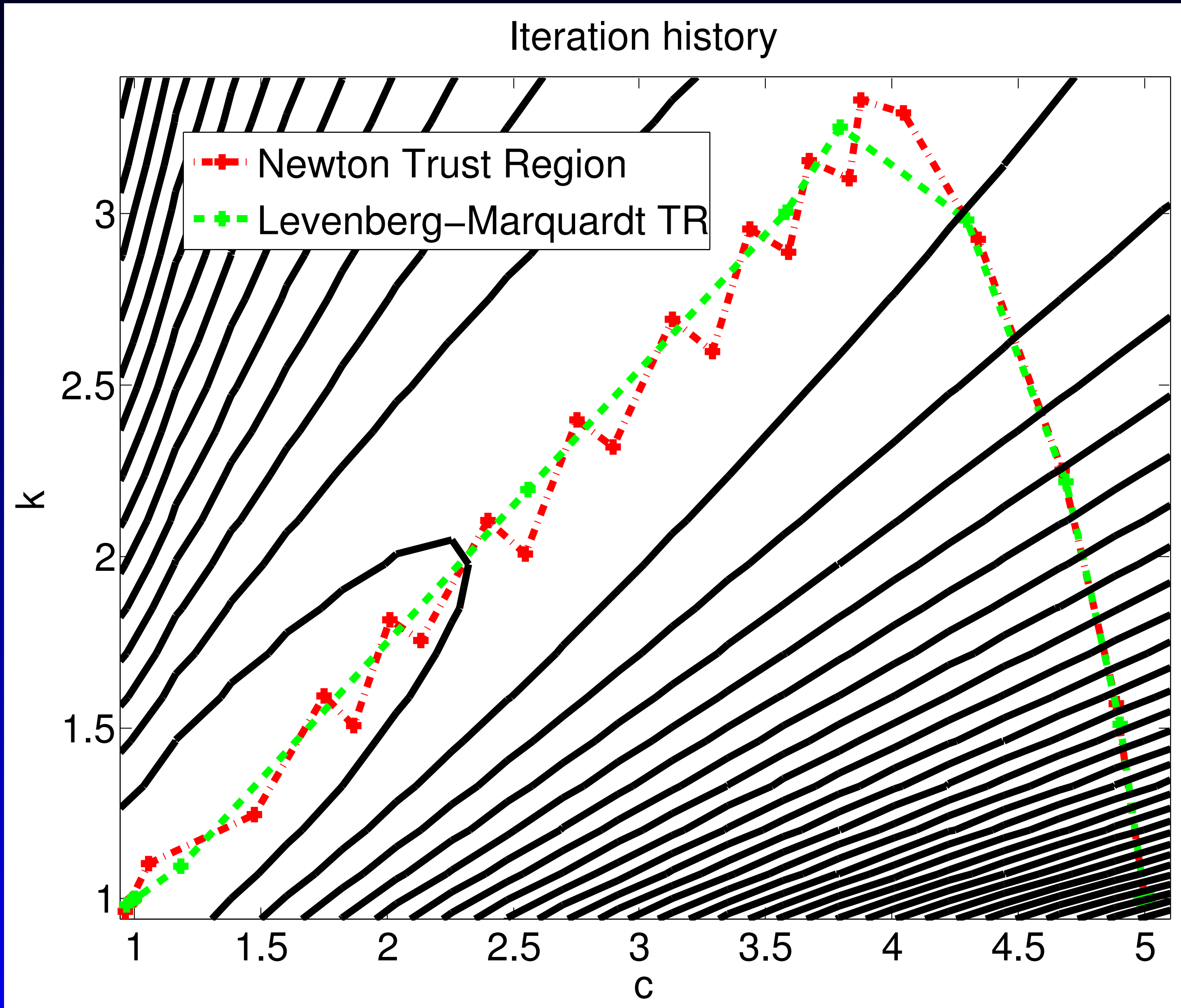
*if and only if there is some $\nu \geq 0$ such that*

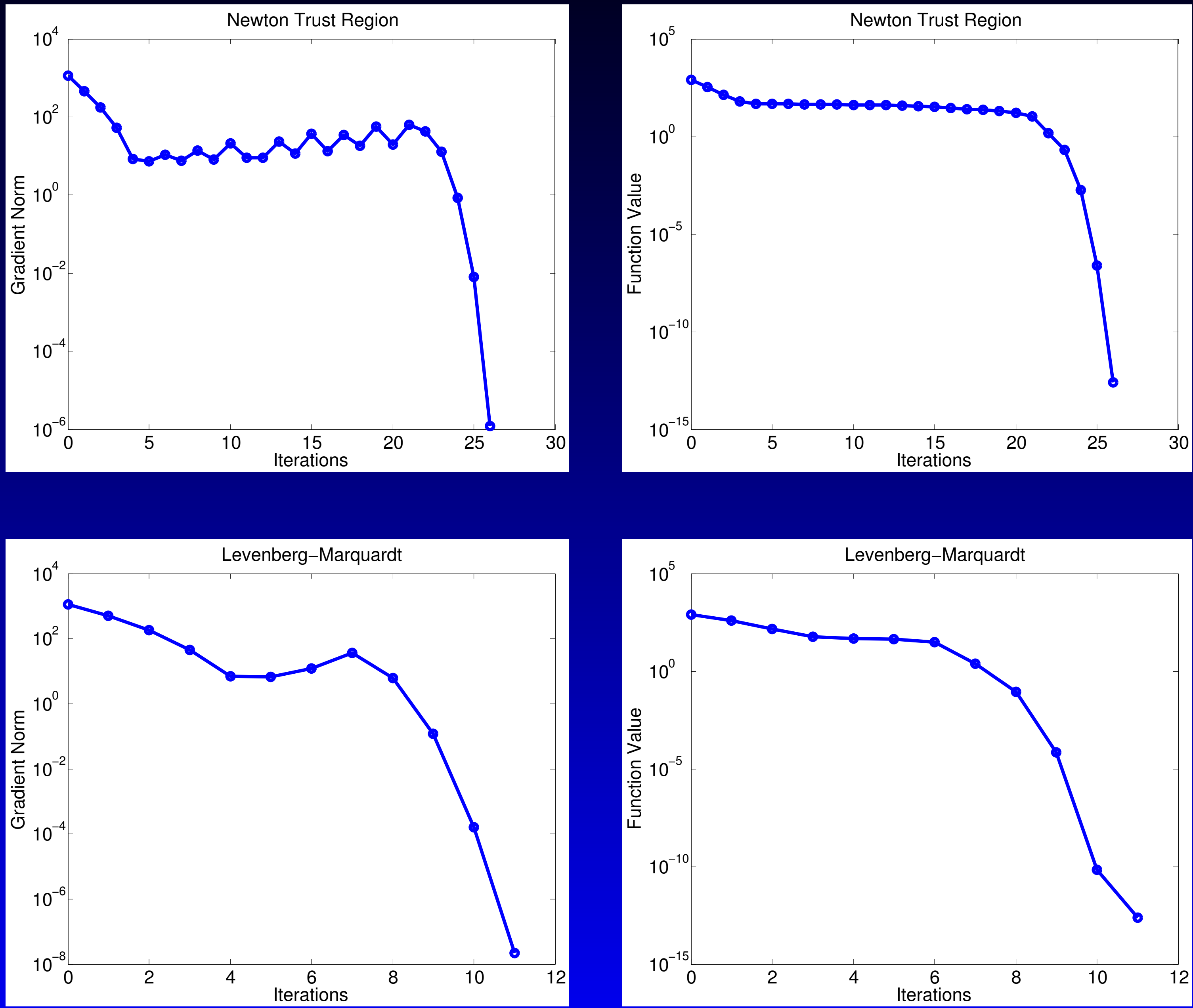
$$(A + \nu I)s = -g$$

*and either $\nu = 0$ or $\|s\| = \Delta$.*

# LM as a TRM

- Instead of controlling $\Delta$ in response to $\mu = ared/pred$, adjust $\nu$.
- Start with $\nu = \nu_0$ and compute $x_t = x_k + s^{LM}$.
- If $\mu = ared/pred$ too small, reject trial and *increase* $\nu$. Recompute trial (only requires a linear solve).
- If $\mu$ sufficiently high, accept trial and possibly *decrease* $\nu$ (maybe to $0$).
- Once trial accepted as an iterate, compute $R$, $f$, $R'$, $\nabla f$ and test $\|\nabla f\|$ for termination.


Iteration history
Newton Trust Region
Levenberg−Marquardt TR
k
c
1
1.5
2
2.5
3
3.5
4
4.5
5
1
1.5
2
2.5
3


Newton Trust Region
Gradient Norm
Iterations
Newton Trust Region
Function Value
Iterations
Levenberg−Marquardt
Gradient Norm
Iterations
Levenberg−Marquardt
Function Value
Iterations

# Summary

- If Gauss-Newton fails, use Levenberg-Marquardt for low-residual nonlinear least squares problems.
  - Achieves global convergence expected of Steepest Descent, but limits to quadratically convergent method near minimizer.
- Use either a trust region or line search to ensure sufficient decrease.
  - Can use trust region with any method that uses quadratic model of $f$.
  - Can only use line search for descent directions.

# References


1. Levenberg, K., “A Method for the Solution of Certain Problems in Least-Squares”, Quarterly Applied Math. 2, pp. 164-168, 1944.
2. Marquardt, D., “An Algorithm for Least-Squares Estimation of Nonlinear Parameters”, SIAM Journal Applied Math., Vol. 11, pp. 431-441, 1963.
3. Moré, J. J., “The Levenberg-Marquardt Algorithm: Implementation and Theory”, Numerical Analysis, ed. G. A. Watson, Lecture Notes in Mathematics 630, Springer Verlag, 1977.
4. Kelley, C. T., “Iterative Methods for Optimization”, Frontiers in Applied Mathematics 18, SIAM, 1999. http://www4.ncsu.edu/~ctk/matlab_darts.html.
5. Wadbro, E., “Additional Lecture Material”, Optimization 1 / MN1, Uppsala Universitet, http://www.it.uu.se/edu/course/homepage/opt1/ht07/.