Gradient-based Methods for Optimization. Part II.

Prof. Nathan L. Gibson

Department of Mathematics

Applied Math and Computation Seminar
October 28, 2011

Summary from Last Time

- Unconstrained Optimization
  - Nonlinear Least Squares
    - Parameter ID Problem

Sample Problem:

$$u'' + cu' + ku = 0; u(0) = u_0; u'(0) = 0 \tag{1}$$

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 ν

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 ν to take more of a SD direction.
- As you get closer to minimizer, decrease ν 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 ν to add regularity.
- As you get closer to minimizer, Hessian will become positive definite. Decrease ν as less regularization is necessary.
- Regularized problem is "nearby problem", want to solve actual problem as soon as feasible.

Outline

- Line Search (Armijo Rule)
  - Damped Gauss-Newton
  - LMA
- Levenberg-Marquardt Parameter
- Polynomial Models
- Trust Region
  - Changing TR Radius
  - Changing LM Parameter

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 λ_k so that $f(x)$ *decreases sufficiently*.
- If we ask simply that

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

  Steepest Descent might not converge.

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

$$\textit{ared} \geq \alpha \textit{ pred},$$

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

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

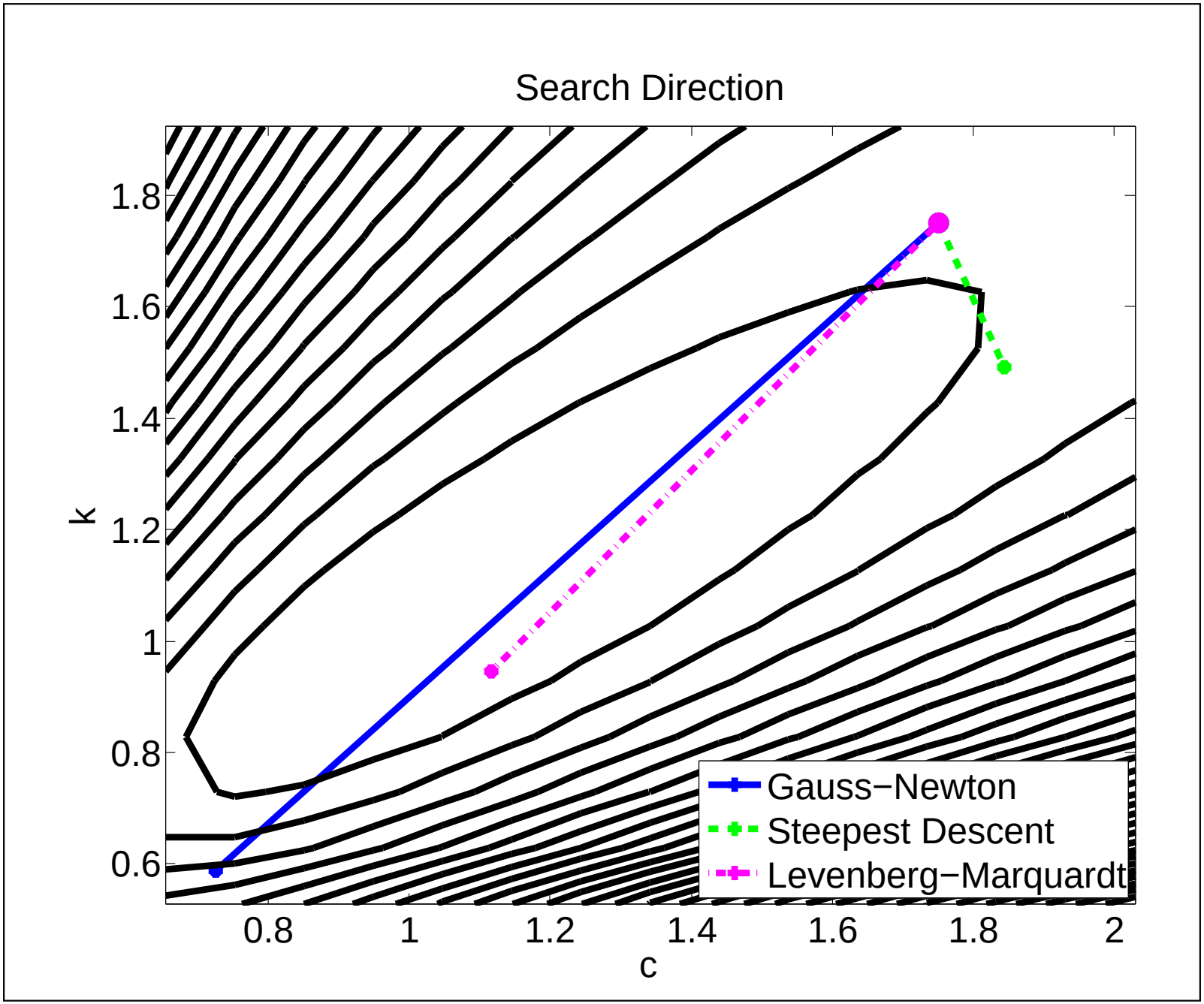
where $\beta \in (0, 1)$ and m is the smallest non-negative integer to guarantee sufficient decrease.

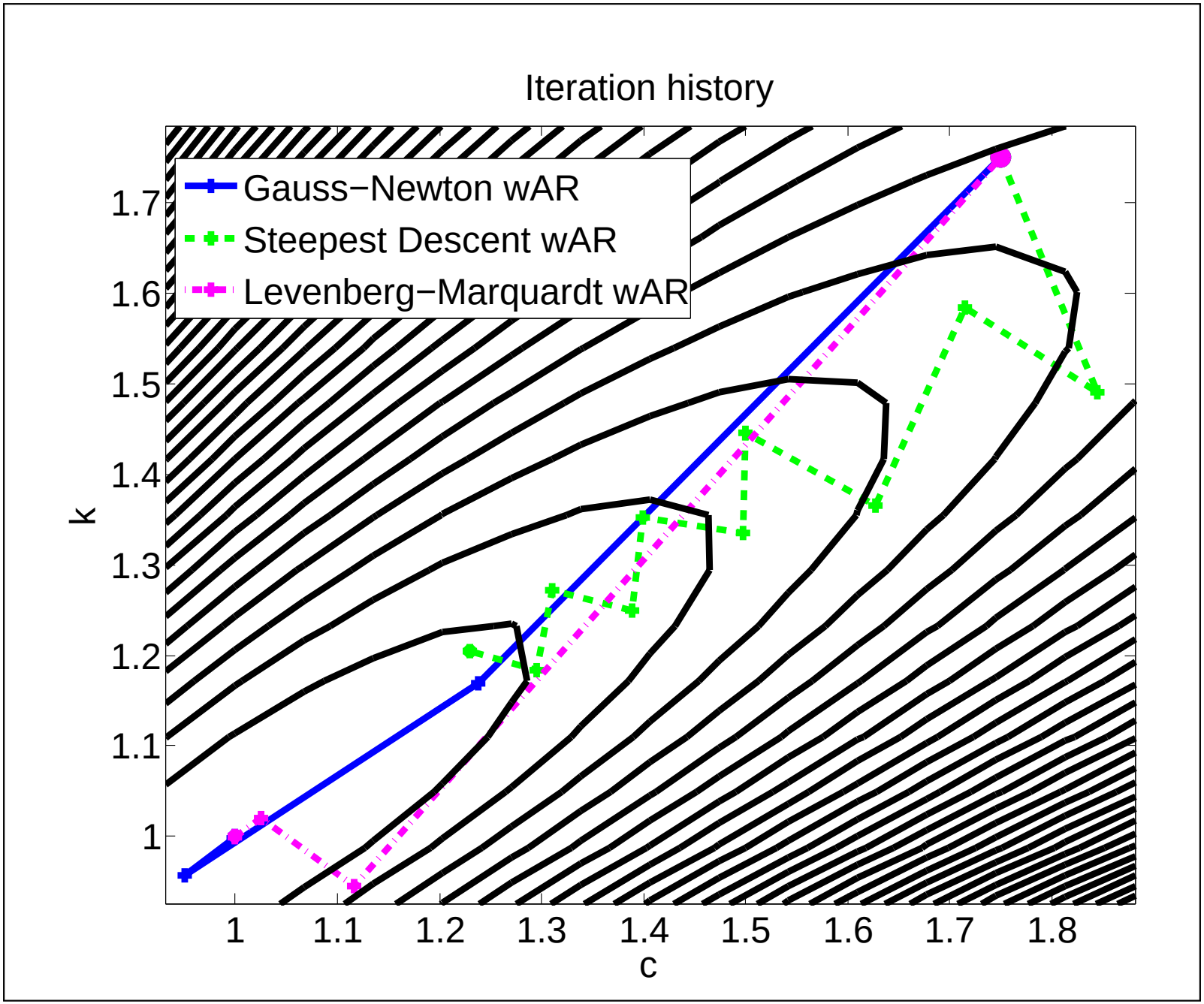
Levenberg-Marquardt-Armijo

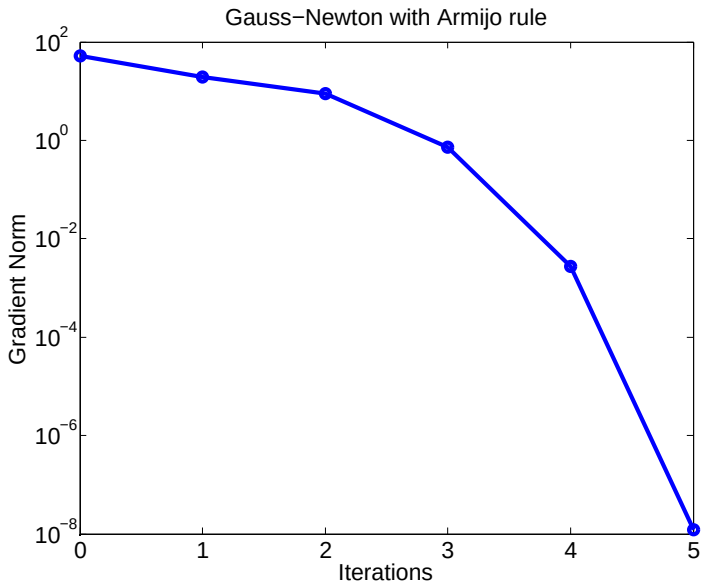
- If $R'(x)$ does not have full column rank, or if the matrix $R'(x)^T R'(x)$ may be ill-conditioned, you should be using Levenberg-Marquardt.
- The LM direction is a descent direction.
- Line search can be applied.
- Can show that if $\nu_k = O(\|R(x_k)\|)$ then LMA converges quadratically for (nice)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

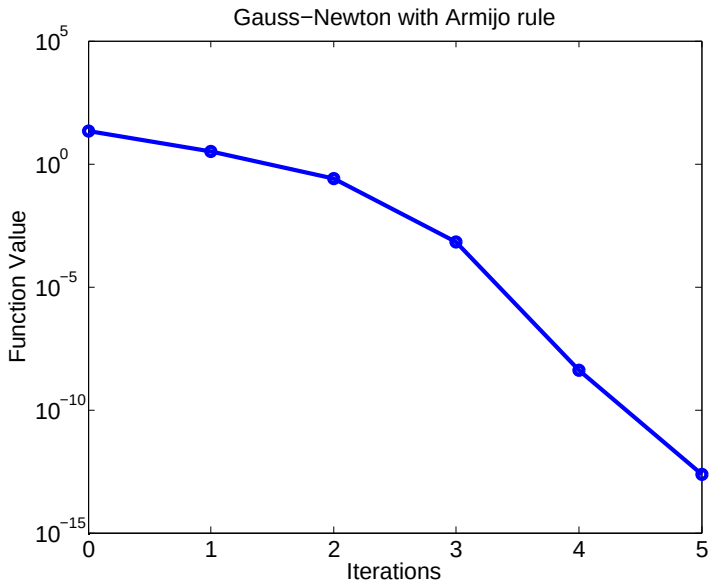
Numerical Example

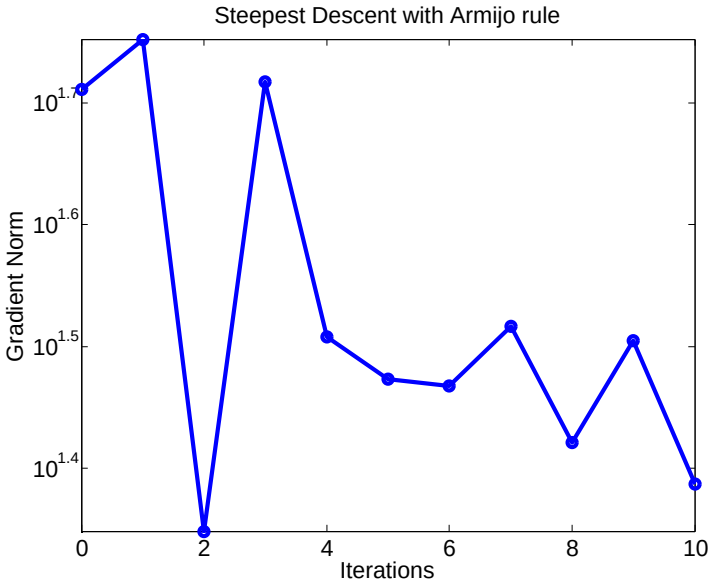
- Recall
$$u'' + cu' + ku = 0; u(0) = u_0; u'(0) = 0.$$
- Let the true parameters be $x^* = [c, k]^T = [1, 1]^T$. Assume we have $M = 100$ data u_j from equally spaced time points on $[0, 10]$.
- We will use the initial iterate $x_0 = [3, 1]^T$ with Steepest Descent, Gauss-Newton and Levenberg-Marquardt methods using the Armijo Rule.

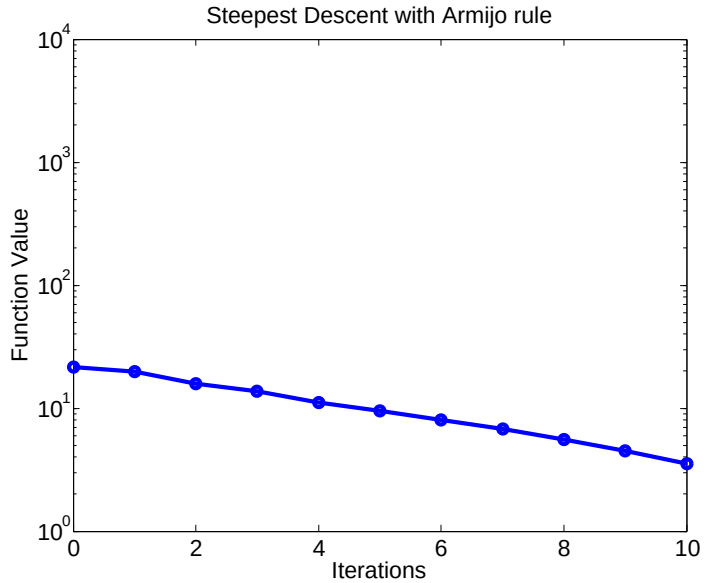

Search Direction
1.8
1.6
1.4
1.2
1
0.8
0.6
k
0.8
1
1.2
1.4
1.6
1.8
2
c
Gauss−Newton
Steepest Descent
Levenberg−Marquardt

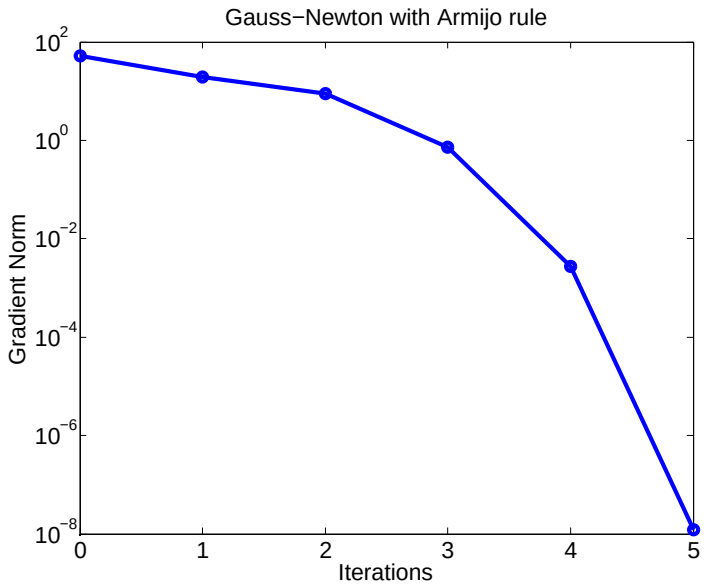

Iteration history
Gauss−Newton wAR
Steepest Descent wAR
Levenberg−Marquardt wAR
k
c
1
1.1
1.2
1.3
1.4
1.5
1.6
1.7
1
1.1
1.2
1.3
1.4
1.5
1.6
1.7
1.8

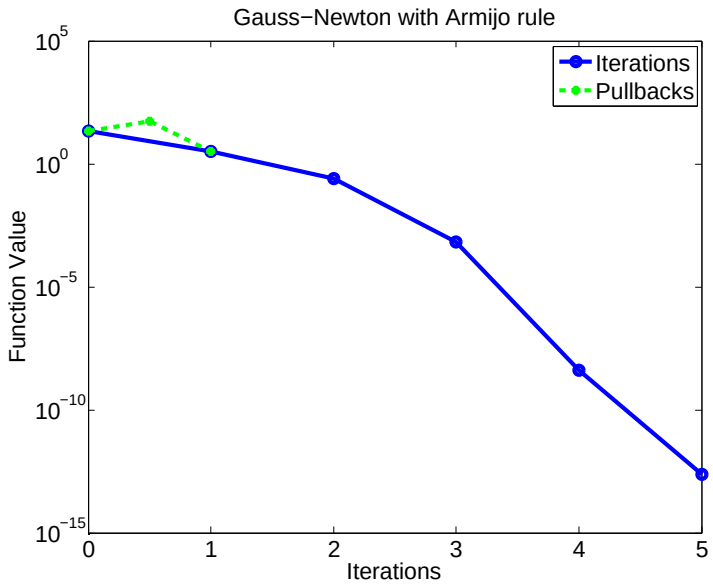

Gauss−Newton with Armijo rule
Gradient Norm
Iterations

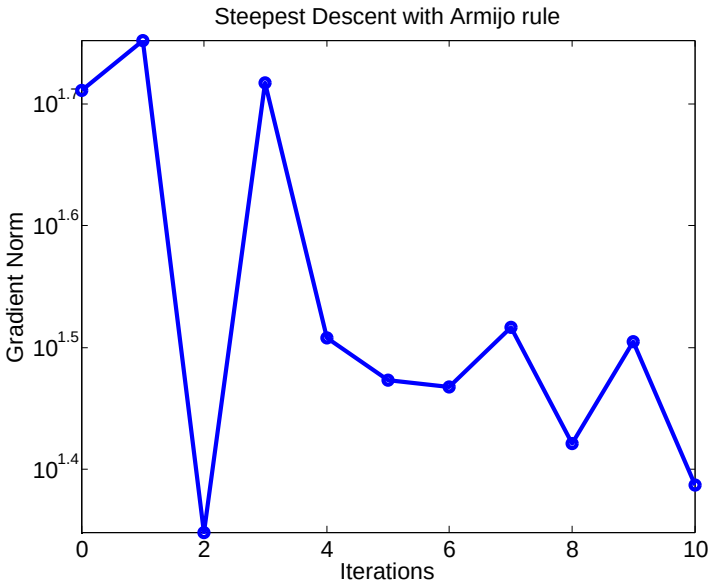

Gauss−Newton with Armijo rule
Function Value
Iterations

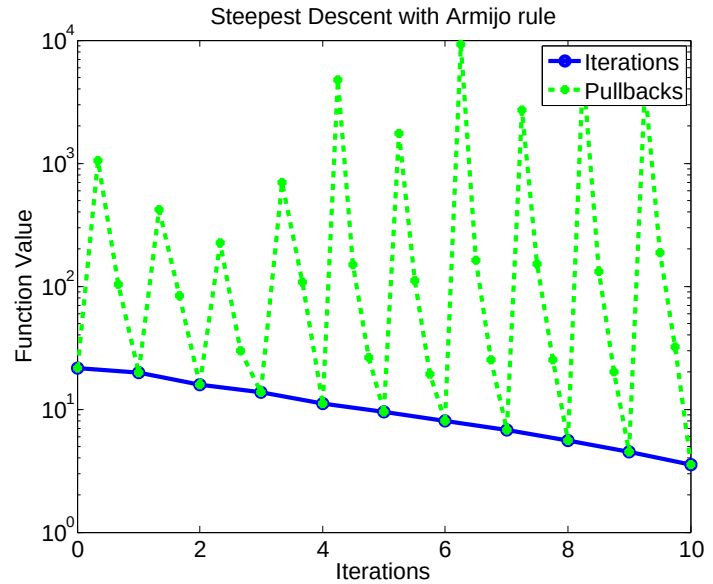

Steepest Descent with Armijo rule
Gradient Norm
Iterations


Steepest Descent with Armijo rule
Function Value
Iterations


Gauss−Newton with Armijo rule
Gradient Norm
Iterations

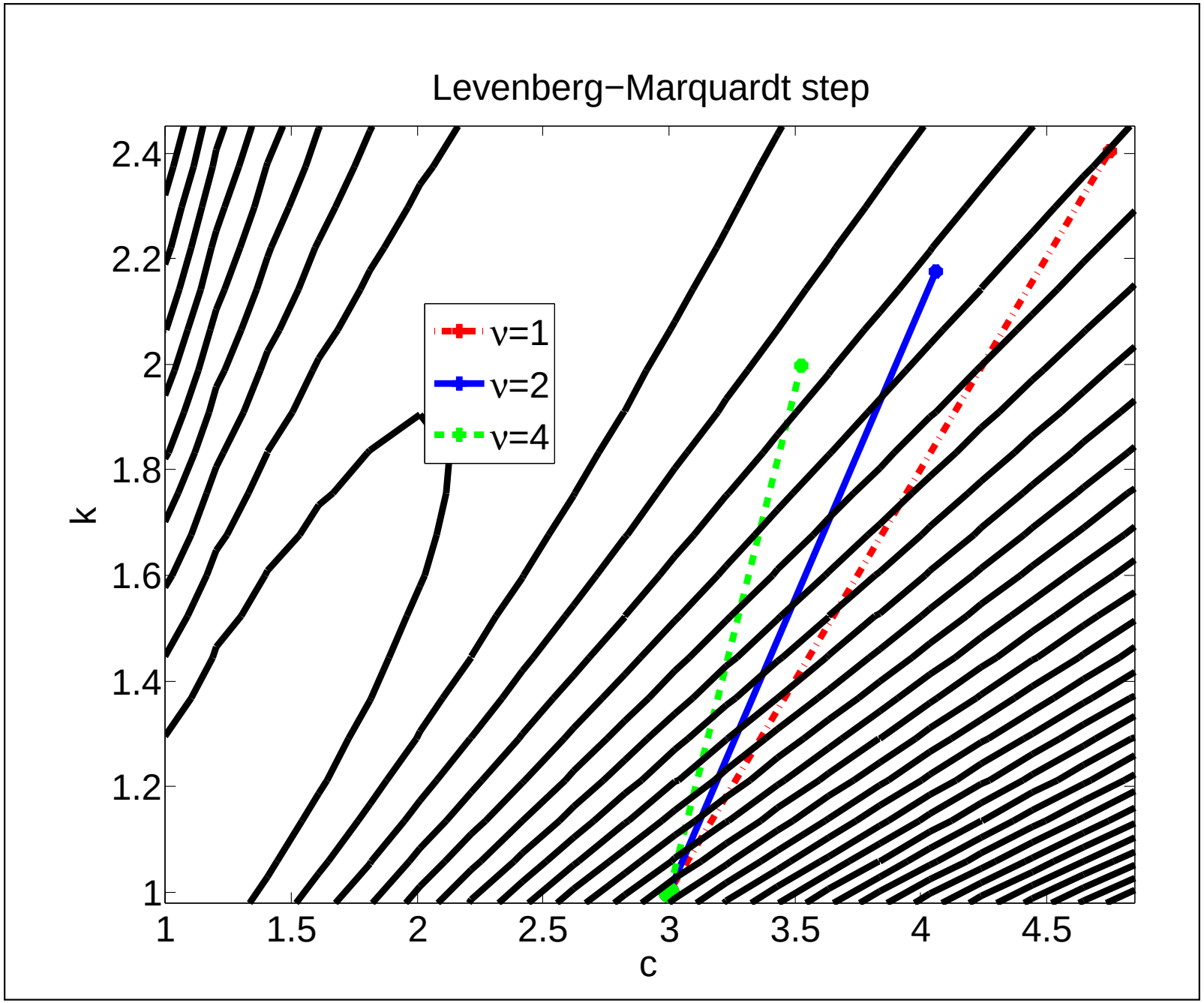

Gauss−Newton with Armijo rule
Iterations
Pullbacks
Function Value
Iterations

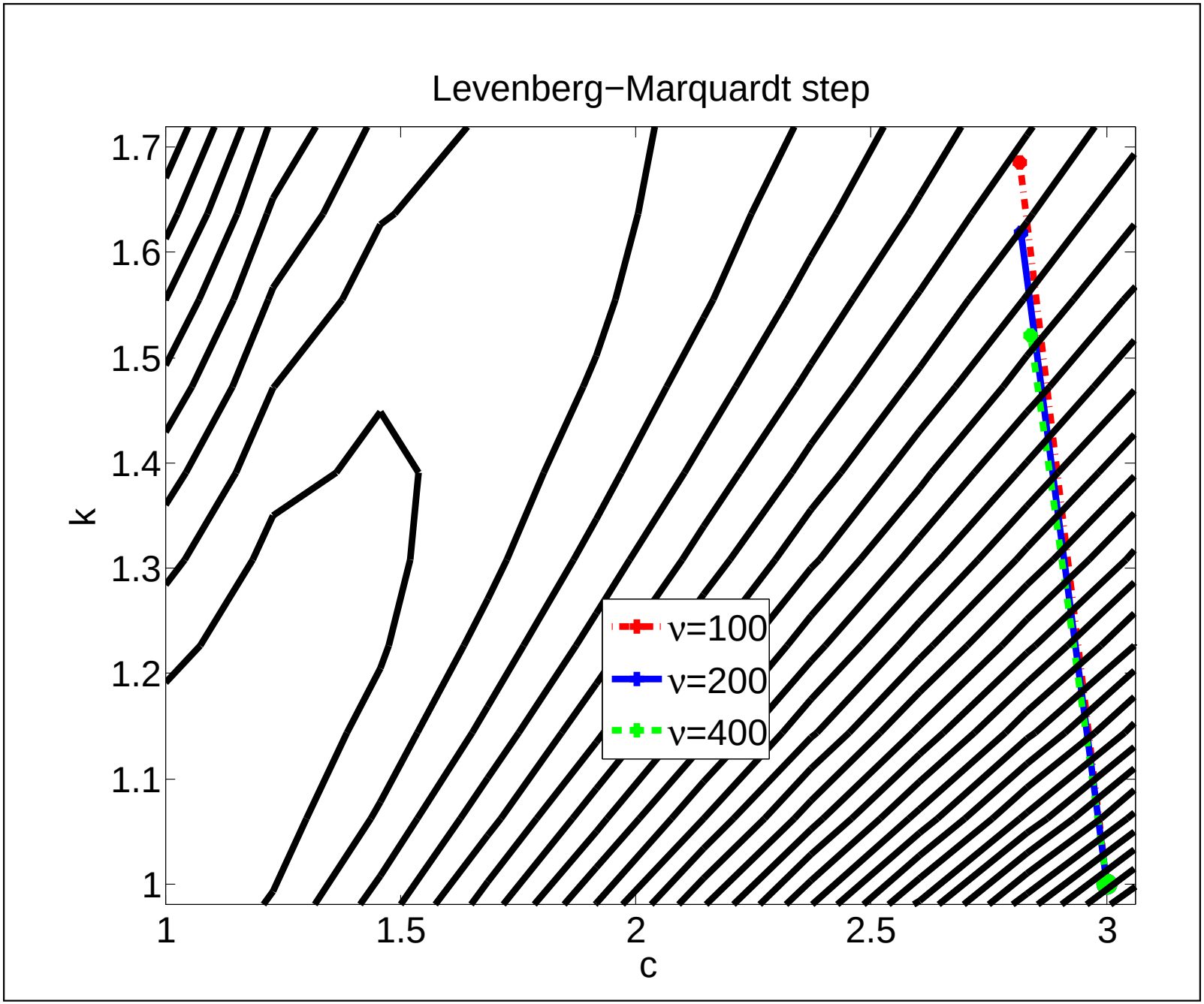

Steepest Descent with Armijo rule
Gradient Norm
Iterations


Steepest Descent with Armijo rule
Iterations
Pullbacks
Function Value
Iterations

Word of Caution for LM

- Note that blindly increasing ν until a sufficient decrease criteria is satisfied is NOT a good idea (nor is it a line search).
- Changing ν changes direction as well as step length.
- Increasing ν does insure your direction is descending.
- But, increasing ν 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 λ which minimizes model of ξ.
- 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 Δ 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 $f(x)$.
- Δ 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.

Changing Trust Region Radius

- 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 μ 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

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

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

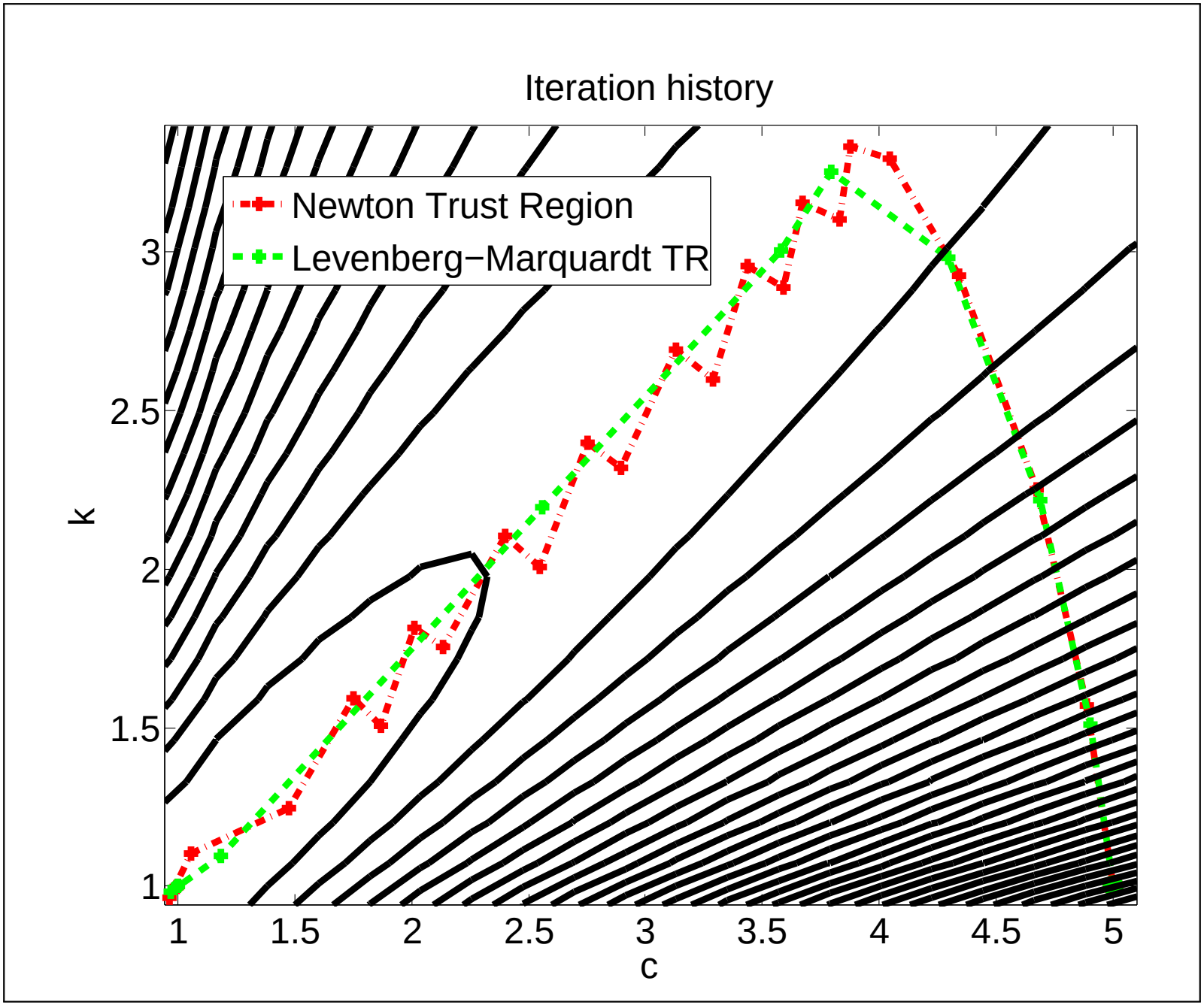
if and only if there is some $\nu \geq 0$ such that

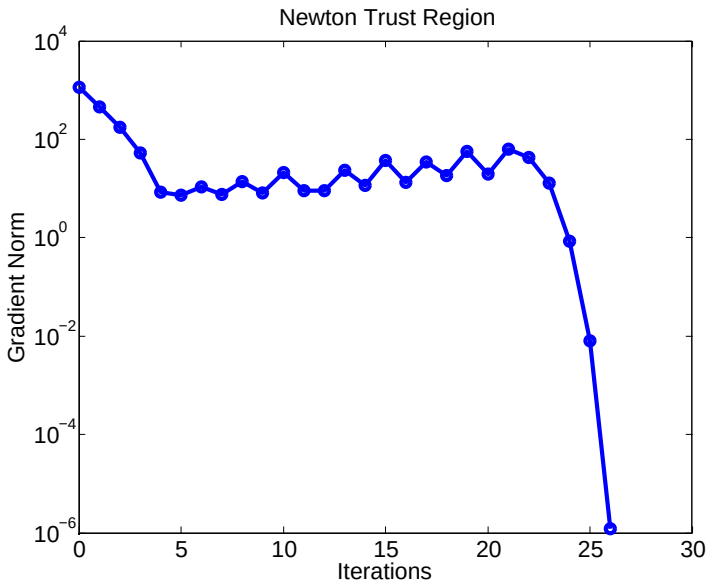
$$(A + \nu I)s = -g$$

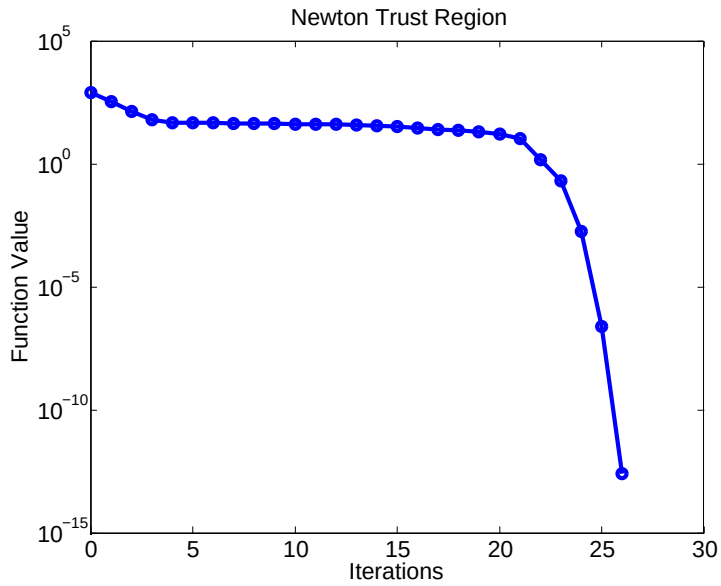
and either $\nu = 0$ or $\|s\| = \Delta$.

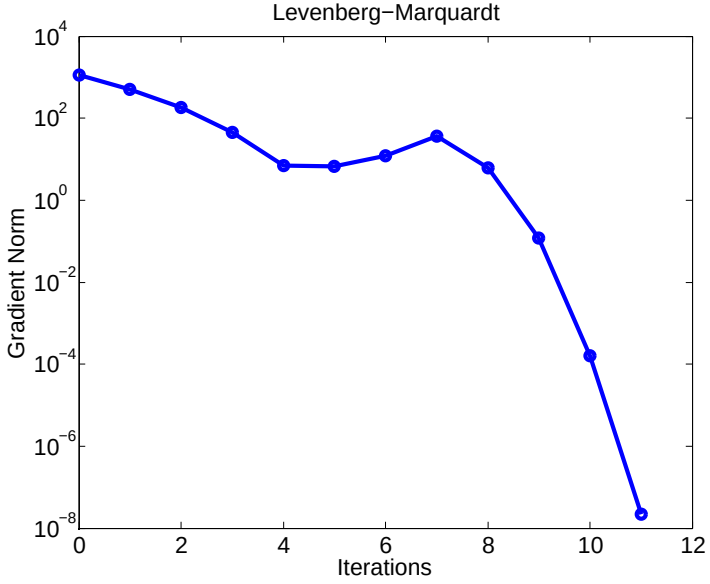
LM as a TRM

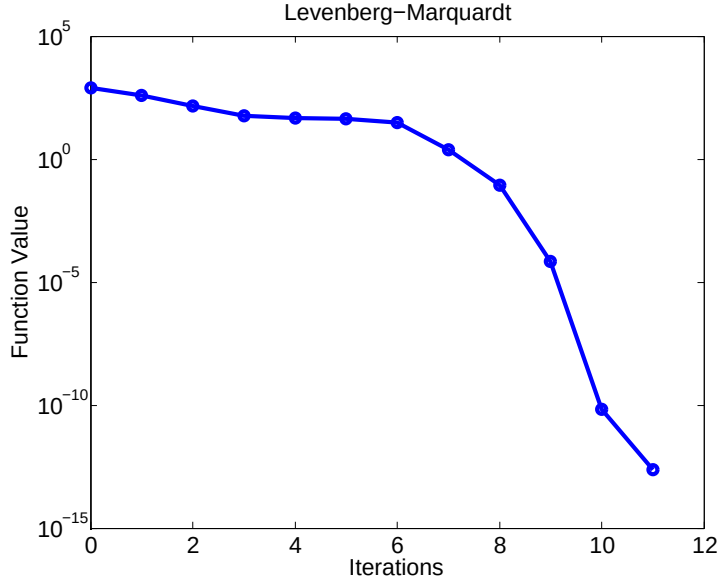
- Instead of controlling Δ in response to $\mu = ared/pred$, adjust ν.
- Start with $\nu = \nu_0$ and compute $x_t = x_k + s^{LM}$.
- If $\mu = ared/pred$ too small, reject trial and *increase* ν. Recompute trial (only requires a linear solve).
- If μ sufficiently high, accept trial and possibly *decrease*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


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/.

Consider $A \in \mathbb{R}^{M \times N}$ and $b \in \mathbb{R}^M$, we wish to find $x \in \mathbb{R}^N$ such that

$$Ax = b.$$

In the case when $M = N$ and A^{-1} exists, the unique solution is given by

$$x = A^{-1}b.$$

For all other cases, if A is full rank, a solution is given by

$$x = A^{+}b$$

where $A^{+} = (A^T A)^{-1} A^T$ is the (Moore-Penrose) psuedoinverse of A. This solution is known as the (linear) least squares solution because it minimizes the ℓ_2 distance between the range of A and the RHS b

$$x = argmin \| b - Ax \|_2.$$

Can also be written as the solution to the normal equation

$$A^T Ax = A^T b.$$

Corollary: There exists a unique least squares solution to $Ax = b$ iff A has full rank.
However, there may be (numerical) problems if A is "close" to rank-deficient, i.e., $A^T A$ is close to singular.

Regularization

One can make $A^T A$ well-posed or better conditioned by adding on a well-conditioned matrix, e.g., $\alpha I, \alpha > 0$ (Tikhonov Regularization). Thus we may solve

$$(A^T A + \alpha I)x = A^T b$$

or equivalently

$$x = \mathit{argmin} \|b - Ax\|_2 + \alpha \|x\|_2$$

where we have added a penalty function.
Of course, now we are solving a different (nearby) problem; this is a trade-off between matching the data (b) and prefering a particular type of solution (e.g., minimum norm).

Linear Least Squares with Uncertainty

Consider solving

$$AX = B - N$$

where now X, B, N are random variables with $N \sim \mathcal{N}(\vec{0}, C_N)$ representing additive Gaussian white noise and we expect the solution X to behave $X \sim \mathcal{N}(\vec{0}, C_X)$ (prior distribution). For any given realization of B we wish to find the expected value of X under uncertainty governed by N.

Maximum Likelihood Estimator

The maximum likelihood estimator answers question: "which value of X is most likely to produce the measured data B?"

$$x_{MLE} = argmax p(b|x) = argmax log p(b|x)$$

where

$$p(b|x) = c \exp\left(-\frac{1}{2}(b - Ax)^T C_N^{-1}(b - Ax)\right)$$

and

$$log p(b|x) = -\frac{1}{2}(b - Ax)^T C_N^{-1}(b - Ax) + \tilde{c}$$

The maximum occurs when

$$0 = \frac{d}{dx} logp(b|x) = A^T C_N^{-1}(b - Ax)$$

or

$$A^T C_N^{-1} Ax = A^T C_N^{-1} b.$$

Note that solution does not depend on assumed distribution for X (ignores prior). If we assume that the error i.i.d., $C_N = \sigma_N^2 I$, then

$$A^T Ax = A^T b$$

and we get exactly the normal equations. Thus if you use the least squares solution, you are assuming i.i.d, additive Gaussian white noise.

Weighted Linear Least Squares

If this is not a good assumption, don't use lsq. For instance, if $C_N = \gamma^2 \Gamma$, Γ spd, then x_{MLE} solves

$$A^T \Gamma^{-1} A x = A^T \Gamma^{-1} b$$

or

$$\min_x \|b - Ax\|_\Gamma$$

otherwise known as weighted least squares.

Maximum a Posteriori Estimator

MAP directly answers the question: "given observation b what is the most likely x?" Consider again

$$AX = B - N$$

with $N \sim \mathcal{N}(\vec{0}, C_N)$ and $X \sim \mathcal{N}(\vec{0}, C_X)$ (prior distribution). Applying Bayes' Law

$$p(x|b) = \frac{p(b|x)p(x)}{p(b)}$$

and taking logs on both sides gives

$$logp(x|z) = -\frac{1}{2}(b - Ax)^T C_N^{-1}(b - Ax) - frac12x^T C_X^{-1}x + \tilde{c}.$$

Differentiating wrt x implies x_{MAP} solves

$$(A^T C_N^{-1} A + C_x^{-1})x = A^T C_N^{-1} b.$$

Tikhonov Regularization (Again)

$$(A^T C_N^{-1} A + C_x^{-1})x = A^T C_N^{-1} b.$$

Assuming $C_N = \sigma_N^2 I$ and $C_X = \sigma_X^2 I$, then

$$\left(A^T A + \left(\frac{\sigma_N}{\sigma_X} \right)^2 I \right) x = A^T b$$

which are exactly the Tikhonov regularized normal equations with

$$\alpha = \left(\frac{\sigma_N}{\sigma_X} \right)^2$$

representing a signal-to-noise ratio (trade-off).