Numerical Algorithms (MU4IN910)

Lecture 5: Nonlinear equations

Stef Graillat

Sorbonne Université

Summary of the previous lecture

Algorithms for optimization in dimension $n \geq 2$

- Direct search methods (pattern search and Nelder-Meade)
- Steepest descent algorithm
- Newton algorithm
- Quasi-Newton algorithms
- Nonlinear conjugate gradient algorithm
- Nonlinear least squares algorithm

Goals

1. Newton method in dimension $n \geq 1$
2. Homotopy continuation methods

Bibliography

- **Scientific Computing, An Introductory Survey, Michael .T. Heath, Revised Second Edition, SIAM, 2018** (the lecture is mainly based on this book and on the associated slides)
- **Scientific Computing with Case Studies, Dianne P. O'Leary, SIAM, 2009** (Part of the lecture is based on this book and on the associated slides)
- Numerical Methods, D. Faires and R. Burden, 3rd edition, Brooks/Cole, 2002

Newton method

Nonlinear equations

- Given a function f, we seek value x such that

$$f(x) = 0$$

- A solution x is called a root of the equation or a zero of the function f.

Nonlinear equations (cont'd)

Two important cases:

- Single nonlinear equation in one unknown, where $f : \mathbb{R} \to \mathbb{R}$
  A solution is a scalar x satisfying $f(x) = 0$

- System of n nonlinear equations in n unknowns, where $\mathbf{f} : \mathbb{R}^n \to \mathbb{R}^n$
  A solution is a vector x for which $\mathbf{f}(x) = 0$

Examples of nonlinear equations

- Examples of nonlinear equations in dimension 1

$$x^2 - 4\sin(x) = 0$$

for which $x = 1.9$ is an approximate solution

- Examples of a system of nonlinear equations in dimension 2

$$\begin{aligned} x_1^2 - x_2 + 0.25 &= 0 \\ -x_1 + x_2^2 + 0.25 &= 0 \end{aligned}$$

for which $x = [0.5\,;\ 0.5]^T$ is a solution

Existence and uniqueness of some solutions

- Existence and uniqueness of solutions are more complicated for nonlinear equations than for linear equations
- If $f : \mathbb{R} \to \mathbb{R}$ is continuous and $\mathrm{sign}(f(a)) \neq \mathrm{sign}(f(b))$ then the Intermediate Value Theorem implies that there exists $x^* \in [a, b]$ such that $f(x^*) = 0$
- There is no simple analog for dimensions $n > 1$.

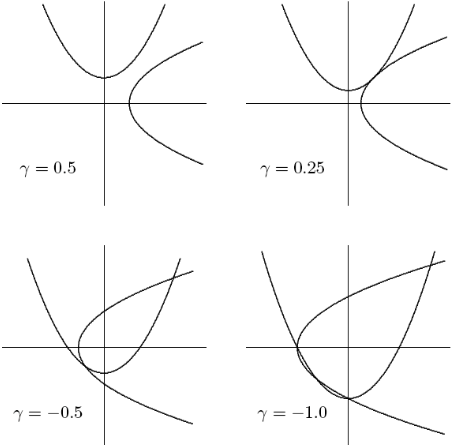
Examples in dimension 1

Nonlinear equations can have any number of solutions:

- $e^x + 1 = 0$ has no solution
- $e^{-x} - x = 0$ has one solution
- $x^2 - 4\sin(x) = 0$ has 2 solutions
- $x^3 + 6x^2 + 11 - 6 = 0$ a 3 solutions
- $\sin(x) = 0$ has infinitely many solutions

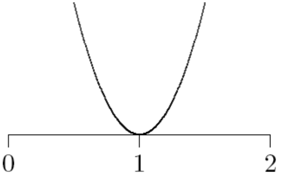
Example in dimension 2

$$\begin{aligned} x_1^2 - x_2 + \gamma &= 0 \\ -x_1 + x_2^2 + \gamma &= 0 \end{aligned}$$

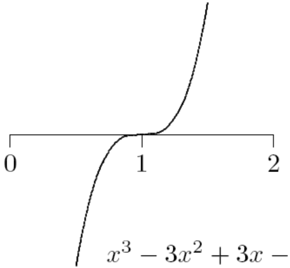


Multiplicity

- If $f(x^*) = f'(x^*) = f''(x^*) = \cdots = f^{(m-1)}(x^*) = 0$ but $f^{(m)}(x^*) \neq 0$, then we say that the root x^* has multiplicity m



$x^2 - 2x + 1$



$x^3 - 3x^2 + 3x - 1$

- If $m = 1$, we say that x^* is a simple root

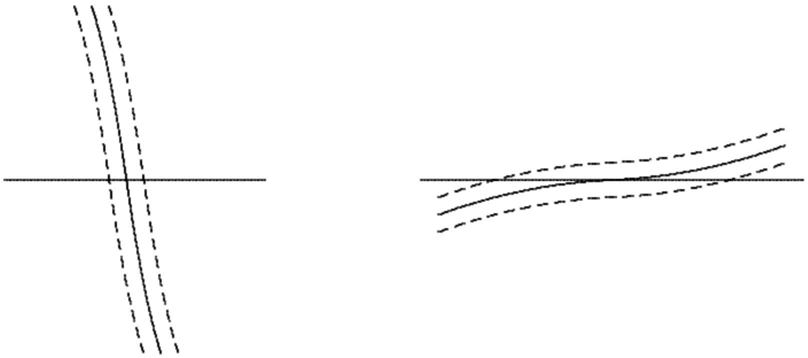
Sensitivity and conditioning

- The conditioning of root finding problem is opposite to that for evaluating function
- The absolute condition number of root finding problem for root x^* of $f : \mathbb{R} \to \mathbb{R}$ is $1/|f'(x^*)|$
- A root is ill-conditioned if its tangent line is nearly horizontal
- In particular, a multiple root is ill-conditioned
- Absolute condition number of root finding problem for root x^* of $f : \mathbb{R}^n \to \mathbb{R}^n$ is $\|J_f^{-1}(x^*)\|$ where J_f is the Jacobian of f,

$$J_f(x) = \{\partial f_i(x)/\partial x_j\}$$

- A root is ill-conditioned if the Jacobian matrix is nearly singular

Sensitivity and conditioning (cont'd)



Sensitivity and conditioning (cont'd)

- What do we mean by approximate solution $\widehat{x}$ to nonlinear system?

$$\|f(\widehat{x})\| \approx 0 \qquad \text{or} \qquad \|\widehat{x} - x^*\| \approx 0?$$

- The first corresponds to "small residual", the second measures the closeness to the true solution
- A small residual implies an accurate solution only if the problem is well-conditioned

Convergence rate

- For a general iterative method, we define the error at iteration k by

$$e_k = x_k - x^*$$

where x_k is an approximation of the solution and x^* is the solution

- The sequence (x_k) is said to converge with a rate r if

$$\lim_{k \to +\infty} \frac{\|e_{k+1}\|}{\|e_k\|^r} = C$$

for a constant $C \geq 0$

Convergence rate (cont'd)

Some interesting special cases

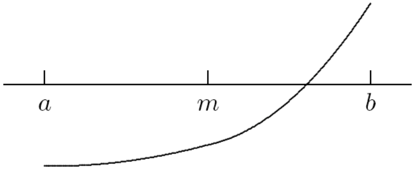
- $r = 1$: linear
- $r > 1$: superlinear
- $r = 2$: quadratic

Convergence rate	Gain in digits per iteration
linear	constant
superlinear	increasing
quadratic	double

Bisection method

The principle of the bisection method is to start from an interval which contains a solution and then divide its length by two until we have isolated the solution with a sufficient accuracy

```
while (b − a) > tol do
    m = a + (b − a)/2
    if sign(f(a)) = sign(f(m)) then
        a = m
    else
        b = m
    end
end
```



Bisection method (cont'd)

- The bisection algorithm converges all the time but slowly
- The rate of convergence is $r = 1$ and $C = 0.5$
- We gain 1 bit of precision at each iteration

Newton's method

- Taylor series of order 1

$$f(x+h) \approx f(x) + hf'(x)$$

This is a linear function in h approximating f in the neighborhood of x

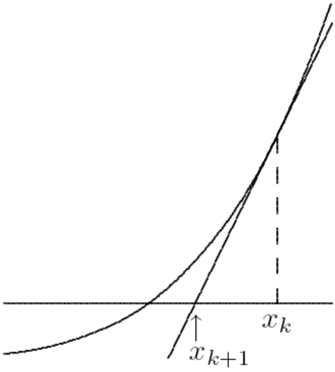
- We replace the nonlinear function f by this linear function whose root is $h = -f(x)/f'(x)$

- The zeros of the function f and the zeros of the linear approximation are in general not identical so we iterate the process

$$x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)}$$

Newton's method (cont'd)

Newton's method approximates a nonlinear function f near x_k by its tangent line at $f(x_k)$.



Convergence of Newton's method

- If the root x^* is simple then the convergence is quadratic ($r = 2$)

- But the iteration must start close enough to the root for convergence

Zeros of univariate polynomials

- Given a polynomial $p(x)$ of degree n, we want to find its n zeros.
- Several approaches exist
  - Use Newton's method to find a zero then deflate and continue
  - Form the companion matrix and compute its eigenvalues
  - Use specific methods to calculate all the roots of a polynomial of a polynomial (Jenkins-Traub, Durand-Kerner, Ehrlich-Aberth, etc.)

Simultaneous methods

- Until the 1960's all known methods for solving polynomials involved finding one root at a time, say z_i, and deflating (i.e. dividing out the factor $z - z_i$). This can lead to problems with increased rounding errors.
- Durand-Kerner method

$$z_{n+1,i} = z_{n,i} - \frac{p(z_{n,i})}{\prod_{j=1,j\neq i}^{m}(z_{n,i} - z_{n,j})}$$

  → quadratic convergence for simple zero (linear convergence for multiple zero) in practice for nearly all starting points
  → compute all the zeros simultaneously
- The Ehrlich-Aberth method combines Newton's method with an implicit deflation strategy. Apply Newton's method to $R_i(z) = \frac{p(z)}{\prod_{j\neq i}(z-z_j)}$ which leads to

$$z_{n+1,i} = z_{n,i} - \frac{1}{\frac{p'(z_{n,i})}{p(z_{n,i})} - \sum_{j=1,j\neq i}^{m} \frac{1}{z_{n,i}-z_{n,j}}}$$

  → cubic convergence for simple zero in practice for nearly all starting points

System of nonlinear equations

The case of the systems of nonlinear equations is much more complex

- a wide variety of possible behavior (determining existence, the number of solutions or a good starting point is more complex)
- The computation time increases rapidly with the dimension of the problem

Newton's Method

- In dimension n, Newton's method is of the form

$$x_{k+1} = x_k - J(x_k)^{-1} f(x_k)$$

where $J(x)$ is the Jacobian matrix of f

- In practice, we do not explicitly compute the inverse of $J(x_k)$ but we solve the linear system

$$J(x_k) s_k = -f(x_k)$$

and we choose

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

Convergence of Newton's method

- The convergence rate of Newton's method is quadratic, provided that the Jacobian matrix is invertible
- But we have to start the iterations close enough of the solution to converge

Cost of Newton's method

The cost per iteration of Newton's method for a dense problem in dimension n is important

- Computing the Jacobian matrix requires n^2 evaluations of functions
- Solving a linear system requires $\mathcal{O}(n^3)$ operations

Newton type methods

- Finite difference method: if J is not available, we can estimate it by finite difference
$$J(x)_{ij} \approx \frac{f_j(x + te_i) - f_j(x)}{t}$$
for t small and e_i the i-th unit vector
- Inexact Newton method: instead of solving the linear system $J(x_k)s_k = -f(x_k)$ exactly, we use an iterative method to obtain an approximate solution (for example the conjugate gradient method)
- Quasi-Newton method: we use an approximation of the Jacobian. For example, the Broyden method
$$B_{k+1} = B_k + \frac{(y - B_k s)s^T}{s^T s}$$
with $y = f(x_{k+1}) - f(x_k)$ and $s = x_{k+1} - x_k$. Then we have $B_{k+1}s = y$ (equation of the secant) and $B_{k+1}v = B_k v$ for $s^T v = 0$

Case of systems of polynomial equations

When one has to solve a system of polynomial equations, there are other very efficient methods

- methods based on the calculation of results
- methods based on the calculation of Gröbner bases

See the book : Mathématiques L3 - Mathématiques appliquées, Jacques-Arthur Weil, Alain Yger, Pearson Education, 2009

Optimization methods

Suppose that the system $f(x) = 0$ is written in the form

$$\begin{cases} f_1(x_1, \ldots, x_n) = 0, \\ f_2(x_1, \ldots, x_n) = 0, \\ \qquad \vdots \\ f_n(x_1, \ldots, x_n) = 0. \end{cases}$$

The system has a solution if and only if the function

$$g(x_1, \ldots, x_n) := \sum_{i=1}^{n} f_i(x_1, \ldots, x_n)^2$$

admits 0 as a minimum

→ one can thus use the optimization algorithms of the previous course to minimize g.

Homotopy continuation methods

Homotopy Methods

- We want to solve $f(x) = 0$
- We know the solution of a simple problem $f_a(x) = 0$ (we can take $f_a(x) = f(x) - f(a)$ for a constant vector a)
- We introduce the problem

$$\rho_a(\lambda, x) = \lambda f(x) + (1-\lambda) f_a(x) = f(x) + (\lambda - 1) f(a)$$

where λ is a real number belonging to the interval $[0;1]$

- The solution of the equation $\rho_a(0, x) = 0$ is known and the solution of the equation $\rho_a(1, x) = 0$ is the solution of our initial problem

Homotopy methods (continued)

Basic algorithm:

Initialize $\lambda = 0$ and x = solution of the equation $f_a(x) = 0$.

while $\lambda < 1$

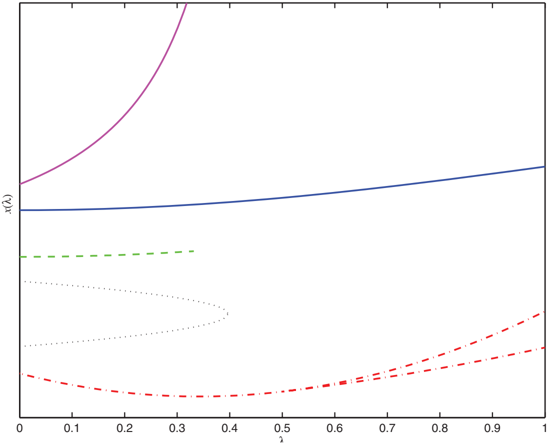
- increase slightly λ
- solve the equation $\rho_a(\lambda, x) = 0$ using the previous solution as a starting point for solving the new problem

end

- We hope that there is a solution for each intermediate problem
- As we slightly modify λ at each step, we also hope that the solution $x(\lambda)$ changes only slightly so that the new equation can be solved easily at each iteration

Homotopy methods (cont'd)

Problems:



- Turning points
- Bifurcation points
- Stopping points (the solution no longer exists for a $\lambda < 1$)
- Divergence to infinity

Homotopy continuation methods (cont'd)

- The difficulty of the method is to construct ρ_a such that we can follow the solution from $\lambda = 0$ to $\lambda = 1$ without losing it
- A function w is said to be transverse to 0 on an open U if for any point $u \in U$ such that $w(u) = 0$, the Jacobian matrix of w in u is of full rank
- Parameterized Sard Theorem : Let $U = \mathbb{R}^n \times [0,1[\times \mathbb{R}^n$ and $\rho : U \to \mathbb{R}^n$ of class C^2 transverse to 0 on U. Let a a point of $\mathbb{R}^n$ and $\rho_a(\lambda, z) = \rho(a, \lambda, z)$. Then the function ρ_a is transverse to 0 on $[0,1[\times \mathbb{R}^n$ for almost all a

Homotopy continuation methods (cont'd)

Theorem : If in addition to the previous hypotheses, the equation $\rho_a(0, z) = 0$ has a unique solution z_0. Then for almost all $a \in \mathbb{R}^n$, there exists a curve solution of $\rho_a(\lambda, z) = 0$ starting from $(0, z_0)$ having the following properties following

- The Jacobian matrix of ρ_a has full rank
- The curve γ is smooth, does not intersect itself and does not intersect any of the other solution curves
- The curve has finite length on any compact $[0, 1[\times\mathbb{R}^n$
- Either the curve reaches the hyperplane $\lambda = 1$ or it diverges to infinity

In summary, one can almost always follow the solution from $\lambda = 0$ to $\lambda = 1$

Homotopy continuation methods (cont'd)

First method:

Given a function ρ_a, let $\lambda = 0$ and $\widehat{x} = a$. Therefore $\rho_a(\lambda, \widehat{x}) = 0$.

Until $\lambda = 1$ do

- Increase slightly λ.
- Solve the equation $\rho_a(\lambda, x) = 0$ using an algorithm (for example Newton's method, etc.) taking $\widehat{x}$ as a starting point
- Call this new solution $\widehat{x}$.

end

If the algorithm finishes, then we have calculated $\widehat{x}$ verifying $\rho_a(1, \widehat{x}) = 0$ and therefore $f(\widehat{x}) = 0$

Problem: choice of the stepsize in λ and of the accuracy of the solution for the solver

Homotopy continuation methods (cont'd)

Second method: via the resolution of differential equations

We have

$$\rho_a(\lambda, x) = \lambda f(x) + (1-\lambda) f_a(x) = f(x) + (\lambda - 1) f(a)$$

- The solution x depends on λ. Therefore $\rho_a(\lambda, x(\lambda)) = 0$.
- By differentiating with respect to λ, we obtain

$$\frac{\partial \rho_a(\lambda, x(\lambda))}{\partial \lambda} + \frac{\partial \rho_a(\lambda, x(\lambda))}{\partial x} x'(\lambda) = 0$$

- A calculation shows that

$$\frac{\partial \rho_a(\lambda, x(\lambda))}{\partial \lambda} = f(a) \quad \text{and} \quad \frac{\partial \rho_a(\lambda, x(\lambda))}{\partial x} = J_f(x(\lambda))$$

We thus obtain the system of differential equations

$$J_f(x(\lambda)) x'(\lambda) = -f(a) \text{ with } x(0) = a$$

Conclusion

Newton's method

- Method for calculating the roots of nonlinear equations
- Use in numerical computation but also in computer algebra (will be seen in tutorial)

Homotopy continuation methods

- Global method