Gradient descent with constraints

Vishal Vasan

International Centre for Theoretical Sciences
Tata Institute of Fundamental Research

vishal.vasan@icts.res.in

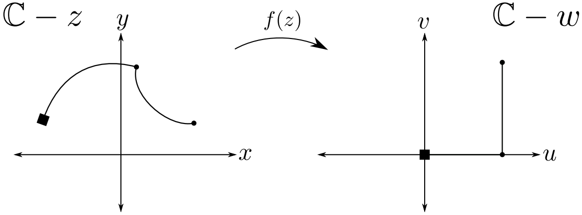
Complex Systems Discussion Meeting, Bengaluru

July 22, 2016

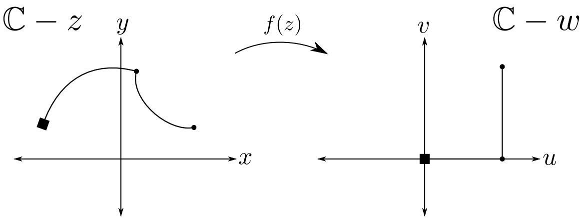
Root finding for analytic functions

Let $f : D \to \mathbb{C}$ be a nontrivial complex-valued function analytic. Find $z_r \in \mathbb{C}$ such that $f(z_r) = 0$

- Lots of literature to review here. Newton's method, Rouché's theorem, Argument principle.
- Basic idea is to construct a differential equation for $z \in D$ such that $z(t) \to z_r$ as $t \to \infty$.



The basic idea



$$\left.\begin{aligned} \frac{dx}{dt} &= u_y = -v_x \\ \frac{dy}{dt} &= -u_x = -v_y \end{aligned}\right\} \Rightarrow \frac{dz}{dt} = \sigma \ \overline{\frac{df}{dz}}$$

- Continuous Newton's method: $\dfrac{dz}{dt} = -f(z)/f'(z)$

Comments about proof of convergence

The algorithm converges if the straight-line path in the $\mathbb{C} - w$ plane corresponds to a continuous curve in the $\mathbb{C} - z$ plane.

- Two canonical paths to connect $w_0 = f(z_0)$ to origin in $\mathbb{C} - w$.
- Little Picard Theorem: entire function omits at most one finite value. At least one of the two paths is in the range.
- Stationary points along the path must be saddle points. If the order of the saddle point is N there may be $N + 1$ roots in a neighborhood of the stationary point.
- Arguments apply with little change to a meromorphic function.
- The boundary of the basins of convergence are smooth.

Polynomial root finding without deflation

- z_m^k be the roots of $f^{(n-m)}(z)$, $f(z)$ n-th order polynomial. μ_m^k is the multiplicity of the root z_m^k.

Lemma

If $f^{(n-m+1)}(z)$ and $f^{(n-m)}(z)$ share a root $z_m^k = z_{m-1}^l = z^*$, then $\mu_{m-1}^l + 1 = \mu_m^k$.

(I) Set $z_1^1 = -\dfrac{a_{n-1}}{na_n}$ and $\mu_1^1 = 1$.

(II) Repeat for $m = 1, \ldots, n-1$:

 (a) Set $l = 1$.

 (b) Repeat for all k: if $f^{(n-m-1)}(z_m^k) = 0$ for any k, set $z_{m+1}^l = z_m^k$, $\mu_{m+1}^l = \mu_m^k + 1$ and increase l by 1.

 (c) If $\sum_l \mu_{m+1}^l < m+1$, then use $\{z_m^k, \mu_m^k\}$, the critical points for $f^{(n-m-1)}$, to define directions of steepest descent/ascent. Follow P_I and/or P_{II} defined from these starting locations to obtain new zeros of $f^{(n-m-1)}$. These are by definition simple zeros, thus $\mu_{m+1}^l = 1$.

(III) z_n^k are the roots of $f(z)$ with multiplicity μ_n^k.

u, v are not action-angle coordinates for

$$\frac{dz}{dt} = \sigma \ \overline{\frac{df}{dz}}$$

Suppose u is conserved, then

$$\frac{dv}{dt} = \frac{\partial v}{\partial x}\frac{dx}{dt} + \frac{\partial v}{\partial y}\frac{dy}{dt} = -\left[\left(\frac{\partial u}{\partial y}\right)^2 + \left(\frac{\partial u}{\partial x}\right)^2\right]$$

Nambu-Poisson systems

- Hamiltonian system

$$\frac{dp}{dt} = -\nabla_p H, \quad \frac{dq}{dt} = \nabla_q H$$

Nambu-Poisson systems

- Hamiltonian system

$$\frac{dp}{dt} = -\nabla_p H, \quad \frac{dq}{dt} = \nabla_q H$$

- Nambu introduced generalisation of Hamiltonian systems

$$\frac{dx}{dt} = \nabla_x H_1 \times \nabla_x H_2, \quad x \in \mathbb{R}^3$$

Nambu-Poisson systems

- Hamiltonian system

$$\frac{dp}{dt} = -\nabla_p H, \quad \frac{dq}{dt} = \nabla_q H$$

- Nambu introduced generalisation of Hamiltonian systems

$$\frac{dx}{dt} = \nabla_x H_1 \times \nabla_x H_2, \quad x \in \mathbb{R}^3$$

- General Nambu-Poisson system

$$\frac{dx}{dt} = S[\nabla_x H_1, \nabla_x H_2, \ldots, \nabla_x H_n], \quad x \in \mathbb{R}^{n+1}$$

  In above example, S is alternating tensor ϵ_{ijk}.

- Many integrable ODEs can be put into Nambu form.

Gradient-Nambu systems

Minimise $g(x, y)$, keep $f(x, y)$ constant

Gradient-Nambu systems

Minimise $g(x, y)$, keep $f(x, y)$ constant

$$\frac{d}{dt}\begin{pmatrix} x \\ y \end{pmatrix} = \nabla f(x, y) \times s(x, y)$$

Gradient-Nambu systems

Minimise $g(x, y)$, keep $f(x, y)$ constant

$$\frac{d}{dt}\begin{pmatrix} x \\ y \end{pmatrix} = \nabla f(x, y) \times s(x, y)$$

to minimise $g(x, y)$

$$\nabla g \cdot \nabla f \times s \leq 0$$

which has a natural "solution"

$$s = \nabla f \times \nabla g$$

Gradient-Nambu systems

Minimise $g(x, y)$, keep $f(x, y)$ constant

$$\frac{d}{dt}\begin{pmatrix} x \\ y \end{pmatrix} = \nabla f(x, y) \times s(x, y)$$

to minimise $g(x, y)$

$$\nabla g \cdot \nabla f \times s \leq 0$$

which has a natural "solution"

$$s = \nabla f \times \nabla g$$

Gradient-Nambu

$$\frac{d}{dt}\begin{pmatrix} x \\ y \end{pmatrix} = \nabla f \times (\nabla f \times \nabla g) = S[\nabla f, S[\nabla f, \nabla g]]$$

Gradient-Nambu systems

Minimise $g(x,y)$, keep $f(x,y)$ constant

$$\frac{d}{dt}\begin{pmatrix} x \\ y \end{pmatrix} = \nabla f(x,y) \times s(x,y)$$

to minimise $g(x,y)$

$$\nabla g \cdot \nabla f \times s \leq 0$$

which has a natural “solution”

$$s = \nabla f \times \nabla g$$

Gradient-Nambu

$$\frac{d}{dt}\begin{pmatrix} x \\ y \end{pmatrix} = \nabla f \times (\nabla f \times \nabla g) = S[\nabla f, S[\nabla f, \nabla g]]$$

Example: $f = u, g = v$ then $\nabla f \times \nabla g = u_x v_y - u_y v_x$ and

$$\frac{d}{dt}\begin{pmatrix} x \\ y \end{pmatrix} = (u_x v_y - u_y v_x)\begin{pmatrix} u_x \\ u_y \end{pmatrix}$$

Gradient-Nambu systems

Minimise $g(x, y, z)$, keep $f_1(x, y, z), f_2(x, y, z)$ constant

Gradient-Nambu systems

Minimise $g(x, y, z)$, keep $f_1(x, y, z), f_2(x, y, z)$ constant

$$\frac{d}{dt}\begin{pmatrix} x \\ y \\ z \end{pmatrix} = S[\nabla f_1, \nabla f_2, S[\nabla f_1, \nabla f_2, \nabla g]]$$

Gradient-Nambu systems

Minimise $g(x, y, z)$, keep $f_1(x, y, z)$, $f_2(x, y, z)$ constant

$$\frac{d}{dt}\begin{pmatrix} x \\ y \\ z \end{pmatrix} = S[\nabla f_1, \nabla f_2, S[\nabla f_1, \nabla f_2, \nabla g]]$$

Higher dimensional generalisations possible.

Gradient-Nambu systems

Minimise $g(x, y, z)$, keep $f_1(x, y, z), f_2(x, y, z)$ constant

$$\frac{d}{dt}\begin{pmatrix} x \\ y \\ z \end{pmatrix} = S[\nabla f_1, \nabla f_2, S[\nabla f_1, \nabla f_2, \nabla g]]$$

Higher dimensional generalisations possible. Easier with tensor notation

Gradient-Nambu systems

Minimise $g(x, y, z)$, keep $f_1(x, y, z), f_2(x, y, z)$ constant

$$\frac{d}{dt}\begin{pmatrix} x \\ y \\ z \end{pmatrix} = S[\nabla f_1, \nabla f_2, S[\nabla f_1, \nabla f_2, \nabla g]]$$

Higher dimensional generalisations possible. Easier with tensor notation :(

Gradient-Nambu systems

Minimise $g(x,y,z)$, keep $f_1(x,y,z), f_2(x,y,z)$ constant

$$\frac{d}{dt}\begin{pmatrix} x \\ y \\ z \end{pmatrix} = S[\nabla f_1, \nabla f_2, S[\nabla f_1, \nabla f_2, \nabla g]]$$

Higher dimensional generalisations possible. Easier with tensor notation :(

(Convex) Dual Gradient-Nambu

Minimise $tf_1 + (1-t)f_2$, keep g constant

$$\frac{d}{dt}\begin{pmatrix} x \\ y \\ z \end{pmatrix} = S[\nabla g, :, S[\nabla g, t\nabla f_1 + (1-t)\nabla f_2, :]]$$

Applications of Gradient Nambu system

- Let $x \in \mathbb{R}^n$
$$\frac{dx}{dt} = f(x), \quad f : \mathbb{R}^n \to \mathbb{R}^n$$
Projection onto orthogonal space
$$I - ff^T/(|f|^2)$$
which has $n-1$ independent columns h_j
- $$\frac{dx}{dt} = S[h_1, h_2, \ldots, h_{n-1}]$$

Applications of Gradient Nambu system

- Let $x \in \mathbb{R}^n$
$$\frac{dx}{dt} = f(x), \quad f : \mathbb{R}^n \to \mathbb{R}^n$$
Projection onto orthogonal space
$$I - ff^T/(|f|^2)$$
which has $n-1$ independent columns h_j
- $$\frac{dx}{dt} = S[h_1, h_2, \ldots, h_{n-1}]$$
- $$\sum \alpha_j h_j = \nabla H$$

Applications of Gradient Nambu system

- Let $x \in \mathbb{R}^n$
$$\frac{dx}{dt} = f(x), \quad f : \mathbb{R}^n \to \mathbb{R}^n$$
Projection onto orthogonal space
$$I - ff^T/(|f|^2)$$
which has $n-1$ independent columns h_j
-
$$\frac{dx}{dt} = S[h_1, h_2, \ldots, h_{n-1}]$$
-
$$\sum \alpha_j h_j = \nabla H$$
- Example:
$$\frac{d}{dt}\begin{pmatrix} p \\ q \end{pmatrix} = \begin{pmatrix} -H_q \\ H_p \end{pmatrix} \Rightarrow \frac{d}{dt}\begin{pmatrix} p \\ q \end{pmatrix} = S[h_1]$$
where
$$h_1 = [H_p^2, H_p H_q]^T \Rightarrow h_1 = \nabla H \Rightarrow H \text{ is conserved}$$

Another application

Differential-algebraic equation

$x \in \mathbb{R}^n, z \in \mathbb{R}^m$

$$\frac{dx}{dt} = f(x,z), \quad g_j(x,z) = 0, j = 1 \dots m$$

Another application

Differential-algebraic equation

$x \in \mathbb{R}^n, z \in \mathbb{R}^m$

$$\frac{dx}{dt} = f(x, z), \quad g_j(x, z) = 0, j = 1 \dots m$$

$\nabla \equiv [D_x, D_z]^T$

$$\frac{d}{dt} \begin{pmatrix} x \\ z \end{pmatrix} = S[\nabla g_1, \dots, \nabla g_m, s_1, \dots, s_n]$$

Another application

Differential-algebraic equation

$x \in \mathbb{R}^n, z \in \mathbb{R}^m$

$$\frac{dx}{dt} = f(x,z), \quad g_j(x,z) = 0, j = 1 \ldots m$$

$\nabla \equiv [D_x, D_z]^T$

$$\frac{d}{dt}\begin{pmatrix} x \\ z \end{pmatrix} = S[\nabla g_1, \ldots, \nabla g_m, s_1, \ldots, s_n]$$

Thank you!