

Homotopy Continuation

Steffen Eibelshäuser*

David Poensgen[†]

April 26, 2021

On the implementation of homotopy continuation in HomCont.

HomCont:

Python software for solving systems of nonlinear equations
by homotopy continuation.

Contents

1	Introduction	2
2	Homotopy Function and Jacobian Matrix	2
2.1	Parameterization	3
3	Predictor-Corrector Procedure	4
	References	8

*Goethe University Frankfurt, eibelshaeuser@econ.uni-frankfurt.de

[†]Goethe University Frankfurt, poensgen@econ.uni-frankfurt.de

1 Introduction

Homotopy continuation¹ denotes a class of numerical techniques to solve high-dimensional, nonlinear systems of equations. The basic idea is the following. Given a complicated problem to solve, first take a similar but simple problem and solve it. Then, gradually transform the simple problem into the complicated problem while holding on to the solution along the way.

Compared to most other numerical methods, homotopy continuation methods have the major advantage of working globally. Iterative Newton-methods for example are only locally convergent, meaning they require a good initial approximation to produce a solution at all. In contrast, homotopy methods arrive at solutions without such a priori knowledge, rendering them an exceptionally powerful tool. In this section, we will briefly sketch the procedure, as a basic understanding is necessary for the following parts of this paper.

The method generally proceeds in two steps: First the formulation of a suitable homotopy function, which implicitly defines a curve from a starting point to the desired solution; and then the numerical traversal along this curve until the solution is obtained. Intuitively, this resembles “bending” the problem until an easy solution is readily available, then reverting it back to the original form, while holding on to the solution.

More concretely, suppose one wants to find a solution \mathbf{x}^* to $F(\mathbf{x}) = \mathbf{0}$, where $F : \mathbb{R}^n \rightarrow \mathbb{R}^n$ is a high-dimensional, nonlinear mapping. One constructs a function $G : \mathbb{R}^n \rightarrow \mathbb{R}^n$, such that a solution $\mathbf{x}_0 \in G^{-1}(\mathbf{0})$ is known or trivially obtained. Then, a homotopy parameter $t \in [0, \bar{t}]$ with $\bar{t} \in (0, \infty]$ ² is introduced to construct a homotopy function $H(\mathbf{x}, t)$, with $H : \mathbb{R}^{n+1} \rightarrow \mathbb{R}^n$, satisfying $H(\mathbf{x}, 0) = G(\mathbf{x})$ and $H(\mathbf{x}, \bar{t}) = F(\mathbf{x})$. If H is constructed properly, it thus offers a continuous transformation of the hard problem $F(\mathbf{x}) = \mathbf{0}$ into the trivial one $G(\mathbf{x}) = \mathbf{0}$ and vice versa. The set of solutions $H^{-1}(\mathbf{0}) = \{(\mathbf{x}, t) | H(\mathbf{x}, t) = \mathbf{0}\}$ then contains a curve connecting the known solution $(\mathbf{x}_0, 0)$ to the desired solution (\mathbf{x}^*, \bar{t}) . Tracing this curve to arrive at the latter is done numerically by predictor-corrector iterations.

2 Homotopy Function and Jacobian Matrix

The system of equations in question can be written as

$$H(\mathbf{y}) = \mathbf{0},$$

where $\mathbf{y} = (\mathbf{x}, t) \in \mathbb{R}^{n+1}$ and $H : \mathbb{R}^{n+1} \rightarrow \mathbb{R}^n$ is a vector-valued homotopy function.

¹For excellent textbook treatments of homotopy methods, see [Zangwill and Garcia \(1981\)](#) and [Allgower and Georg \(1990\)](#).

²The case of $\bar{t} = \infty$ is explicitly included. In this case, path tracing is continued until \mathbf{x} converges.

The corresponding Jacobian matrix $J : \mathbb{R}^{n+1} \rightarrow \mathbb{R}^n \times \mathbb{R}^{n+1}$ is defined by

$$J(\mathbf{y}) = \frac{\partial H(\mathbf{y})}{\partial \mathbf{y}}.$$

Homotopy continuation works best if the Jacobian matrix is explicitly supplied by the user. Alternatively, the Jacobian can be approximated by finite differences schemes, which is typically feasible for small and medium scale problems.

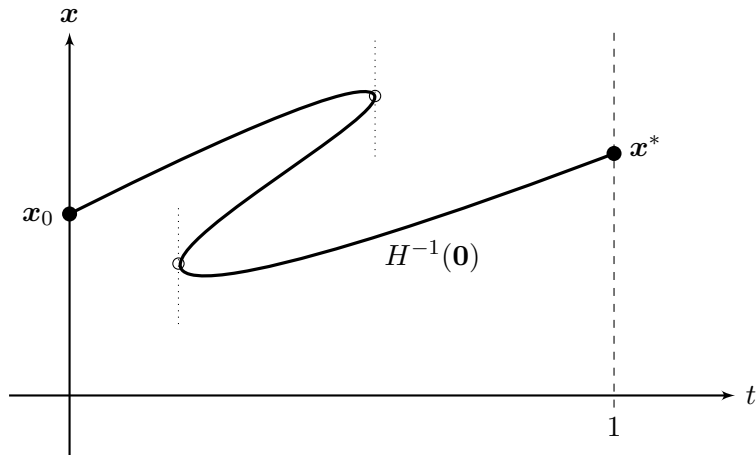
2.1 Parameterization

The homotopy path might have turning points in the sense that the homotopy parameter t is not monotonously increasing along the path, as illustrated in [Figure 1](#). Therefore, it is generally not possible to follow the path by naively increasing t . Instead, it is convenient to parameterize the homotopy function in terms of the path length parameter $\tau \in \mathbb{R}_0^+$ such that $H(\mathbf{x}(\tau), t(\tau)) = \mathbf{0}$. Then,

$$\frac{\partial(\mathbf{x}, t)_k}{\partial \tau} = \eta \cdot (-1)^k \cdot \det(J^{(-k)}(\mathbf{x}, t)) \quad (k \in \{1, \dots, n+1\}),$$

where $J(\mathbf{x}, t) = \frac{\partial H(\mathbf{x}, t)}{\partial(\mathbf{x}, t)}$ denotes the Jacobian matrix $J : \mathbb{R}^{n+1} \rightarrow \mathbb{R}^n \times \mathbb{R}^{n+1}$ of the homotopy function, $J^{(-k)}(\mathbf{x}, t)$ denotes the Jacobian without its k -th column and $\eta \in \mathbb{R}^+$ is normalization factor. For details, see [Zangwill and Garcia \(1981, pp. 25 ff.\)](#).

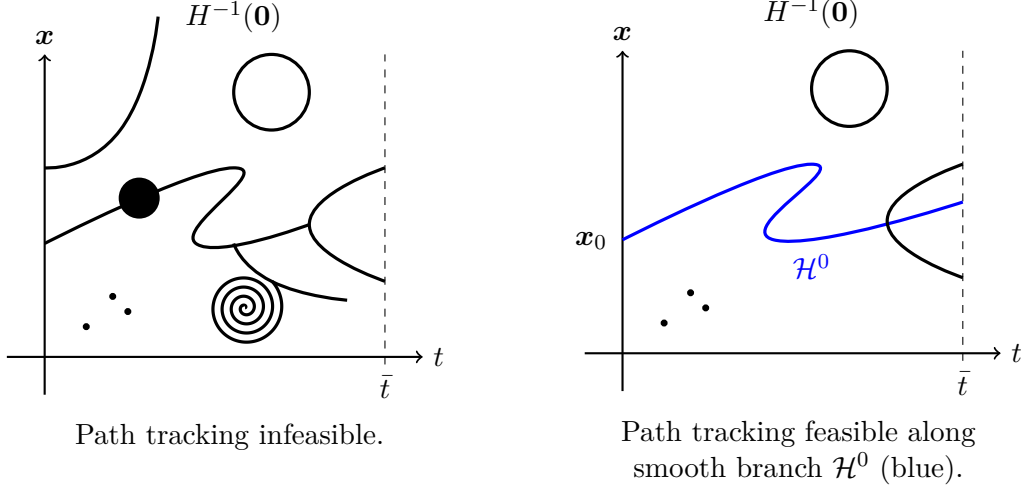
Figure 1: Turning Points of Homotopy Path



In general, the homotopy path $H^{-1}(\mathbf{0})$ is not guaranteed to be as well-behaved as suggested by [Figure 1](#). It might feature multidimensional segments, bifurcations, dead ends or spirals. For path tracking to be well-defined, the homotopy path $H^{-1}(\mathbf{0})$ must include a

smooth branch \mathcal{H}^0 through $(x_0, 0)$ that is almost everywhere one-dimensional, with only isolated transversals of auxiliary path segments. A corresponding illustration is provided in Figure 2.

Figure 2: Possible Shapes of Homotopy Path



Luckily, Sard's Theorem guarantees that *almost all* systems of equations are well-behaved. If the system of interest happens not to be well-behaved, any random perturbation with continuous support will make the system well-behaved with probability one.

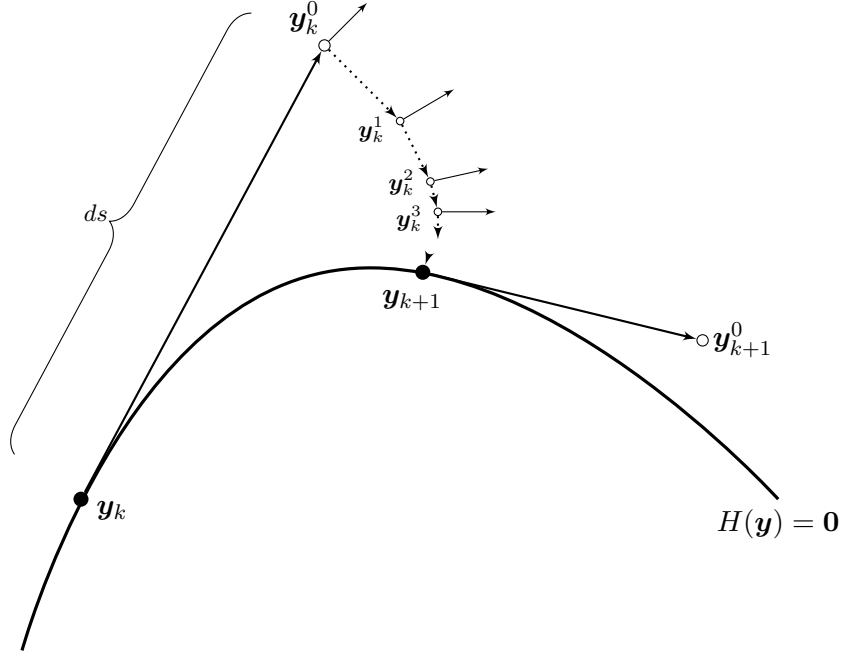
3 Predictor-Corrector Procedure

Predictor-corrector procedures are *the* standard tool to trace differentiable homotopy paths. As the name suggests, these are two-phase procedures, sequentially performing a prediction step and multiple correction steps. In the predictor step, the path at current point \mathbf{y}_k is extrapolated along its tangent with step size ds . Afterwards, the predictor point \mathbf{y}_k^0 is refined by a number of Newton corrector steps. Corrector steps are performed orthogonally to current tangents until a new point \mathbf{y}_{k+1} on the path is reached. Then, the step size is adapted and the two-step procedure is repeated, as illustrated in Figure 3.

Direction. The homotopy path implied by $H(\mathbf{y}) = 0$ is defined up to its direction $\alpha \in \{1, -1\}$. In order to obtain the correct direction for path traversal, α is chosen such that the very first predictor step *increases* t and is held constant thereafter, except in the case of crossing a bifurcation point.

Bifurcation detection. The principal branch of the homotopy path might be crossed by another branch at some point, as illustrated in Figure 4. These so-called simple bi-

Figure 3: Predictor-Corrector Procedure



furcations are singular points of the Jacobian matrix at which the direction of the path may be reversed (Allgower and Georg, 1990, chapter 8). In order to ensure continuation after the bifurcation, simple bifurcation points must be detected and, in case of a reversal of direction, the sign of direction α must be swapped. To detect bifurcations, the angle between consecutive predictor tangents is checked at each step. If the angle is close to 180° and the tangents point in almost opposite direction, the algorithm considers a bifurcation point with reversal of direction crossed. Specifically, the sign of α is swapped if

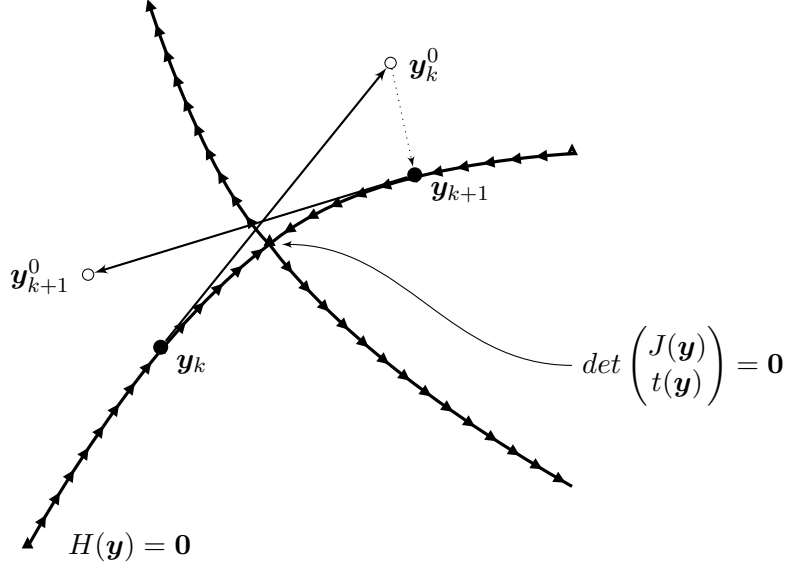
$$[t(\mathbf{y}_k)]^T \cdot t(\mathbf{y}_{k-1}) < \cos(\gamma_{\min})$$

with minimum angle γ_{\min} to classify changes in direction as bifurcation.

Predictor tangent. At each point \mathbf{y}_k , the predictor tangent is computed based on a complete QR decomposition of the transpose $[J(\mathbf{y}_k)]^T$ of the Jacobian at point \mathbf{y}_k . After successful QR decomposition, the tangent is essentially given by the last column of matrix Q , adjusted for the sign of the determinant of matrix R . Specifically, tangent $t(\mathbf{y}_k)$ is computed as

$$t(\mathbf{y}_k) = \alpha \cdot \text{sign}(\det(R)) \cdot Q^{(n+1)}.$$

Figure 4: Simple Bifurcation



Given step size ds and tangent $t(\mathbf{y}_k)$, the predictor point \mathbf{y}_k^0 is given by

$$\mathbf{y}_k^0 = \mathbf{y}_k + ds \cdot t(\mathbf{y}_k).$$

Newton correction. The Newton correction is based on the Moore-Penrose pseudoinverse $[J(\mathbf{y}_k^0)]^+$ of the Jacobian at predictor point \mathbf{y}_k^0 . To be precise, the algorithm uses a Newton-Chord algorithm in which the pseudoinverse is only computed once at the prediction and used for all corrector steps. The pseudoinverse is computed based on a QR decomposition of $[J(\mathbf{y}_k^0)]^T$ as

$$[J(\mathbf{y}_k^0)]^+ = Q \cdot \begin{pmatrix} (R^{(-t)})^{-1} \\ \mathbf{0} \end{pmatrix}$$

where $R^{(-t)}$ denotes matrix R without the row corresponding to differentiation with respect to t within J^T . Given the pseudoinverse, corrector steps l are performed analogously to Newton's method, i.e.

$$\mathbf{y}_k^l = \mathbf{y}_k^{l-1} - [J(\mathbf{y}_k^0)]^+ \cdot H(\mathbf{y}_k^{l-1}),$$

and iterated until either the tracking tolerance H_{tol} is achieved or until failure (see next paragraph). To be conservative, the maximum norm is used to evaluate deviations from

the path, i.e. the correction successfully terminates if

$$\max\{|H(\mathbf{y}_k^l)|\} < H_{\text{tol}}.$$

Newton robustness. In order to ensure safe path traversal, the algorithm imposes a number of robustness requirements on the Newton correction. If one of the robustness criteria fails, i.e. if the convergence of the Newton correction is not ensured, the correction is aborted the predictor step is repeated with a decreased step size.

The correction is considered unsuccessful if either (1) the number of corrector steps L exceeds a threshold, (2) the distance d_l of any corrector step relative to the predictor step size exceeds a threshold or (3) the contraction of consecutive corrector steps, i.e. the ratio $\frac{d_l}{d_{l-1}}$ of distances exceeds a threshold. (4) Finally, following Choi et al. (1996), the algorithm additionally requires that the determinant of the augmented Jacobian does not change too much in the correction. Specifically, the correction is also considered unsuccessful if

$$\left| \frac{\det \begin{pmatrix} J(\mathbf{y}_k^L) \\ t(\mathbf{y}_k) \end{pmatrix}}{\det \begin{pmatrix} J(\mathbf{y}_k) \\ t(\mathbf{y}_k) \end{pmatrix}} \right| > \bar{\Delta}_J \quad \text{or} \quad \left| \frac{\det \begin{pmatrix} J(\mathbf{y}_k^L) \\ t(\mathbf{y}_k) \end{pmatrix}}{\det \begin{pmatrix} J(\mathbf{y}_k) \\ t(\mathbf{y}_k) \end{pmatrix}} \right| < \frac{1}{\bar{\Delta}_J}$$

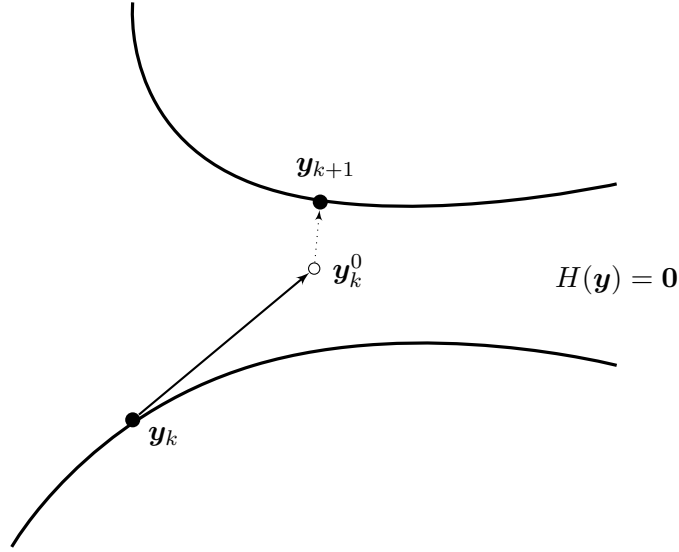
for a given maximum change $\bar{\Delta}_J > 0$. This robustness requirements prevents accidental “path jumping” to a different nearby segment illustrated in Figure 5. When converging to a different path, the Jacobian typically changes so much that the correction is not accepted.

Step size adaption. After each predictor-corrector iteration, the step size is adjusted according to the performance of the Newton correction. In case of unsuccessful correction, the step size is reduced by a deflation factor $f_{\text{defl}} < 1$. In case of successful but slow correction, the step size is held constant. And in case of successful and fast correction, the step size is increased by an inflation factor $f_{\text{infl}} > 1$. The speed of correction is considered fast if less than ten corrector steps were performed until convergence.

Convergence. Two cases are distinguished. First, if $\bar{t} < \infty$, homotopy continuation is considered converged when $|t - \bar{t}| < t_{\text{tol}}$. Second, if $\bar{t} = \infty$, homotopy continuation is considered converged when the change in \mathbf{x} between consecutive predictor-corrector steps relative to the step size falls below a tolerance level x_{tol} . To be conservative, the maximum norm is used to measure differences in \mathbf{x} . For further generality, one can define a transformation function $f : \mathbb{R}^{n+1} \rightarrow \mathbb{R}^m$ and consider changes in $f(\mathbf{x})$ for convergence.³

³This is particularly useful if the homotopy function is parameterized in for example logarithms of \mathbf{x} or some other unbounded transformation.

Figure 5: Path Jumping



Specifically, the algorithm reports convergence if

$$\max \left\{ \frac{|f(\mathbf{x})_k - f(\mathbf{x})_{k-1}|}{ds} \right\} < x_{\text{tol}}.$$

In order not to impede performance, the algorithm checks for convergence in \mathbf{x} only if the current step size is equal to the maximum step size, indicating the path has been relatively smooth for many consecutive steps.

References

- ALLGOWER, E. L. AND K. GEORG (1990): *Numerical Continuation Methods: An Introduction*, New York: Springer.
- CHOI, S., D. A. HARNEY, AND N. BOOK (1996): “A Robust Path Tracking Algorithm for Homotopy Continuation,” *Computers & Chemical Engineering*, 20, 647–655.
- ZANGWILL, W. I. AND C. B. GARCIA (1981): *Pathways to Solutions, Fixed Points, and Equilibria*, Upper Saddle River, New Jersey: Prentice-Hall.