GridCal

Santiago Peñate Vera

GRIDCAL

Research oriented power systems software.

Started writing this document in Madrid the 9th of October of 2016

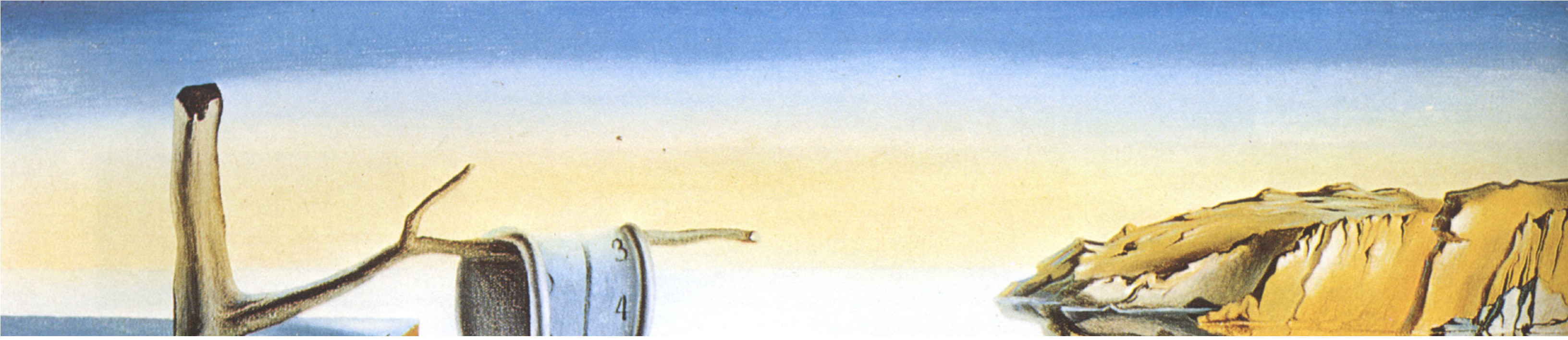


Table of content

1 Introduction 4

1.1 Motivation 4

2 Structure 5

2.1 title 5

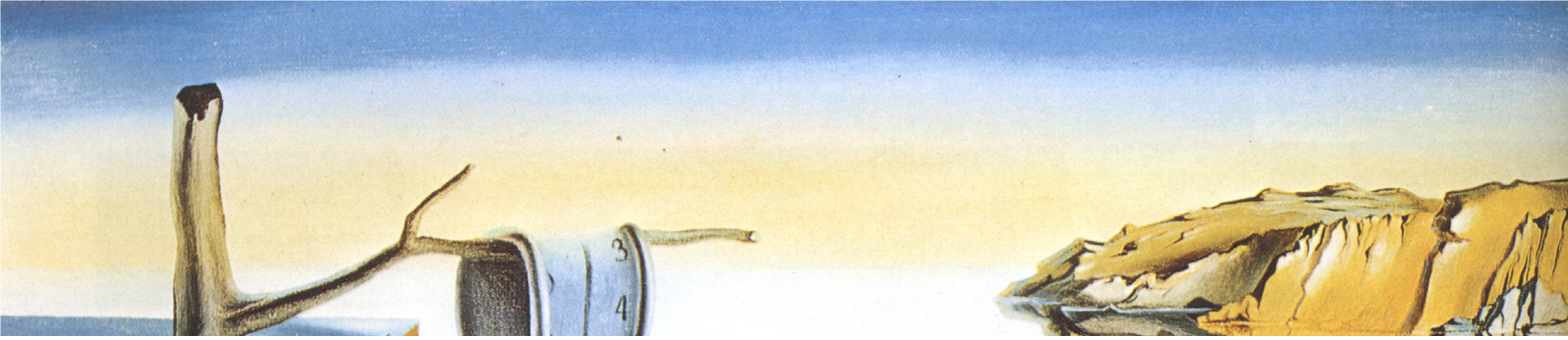
3 Power flow methods 6

3.1 Newton-Raphson-Iwamoto 6

3.2 Holomorphic Embedding (ASU) 7

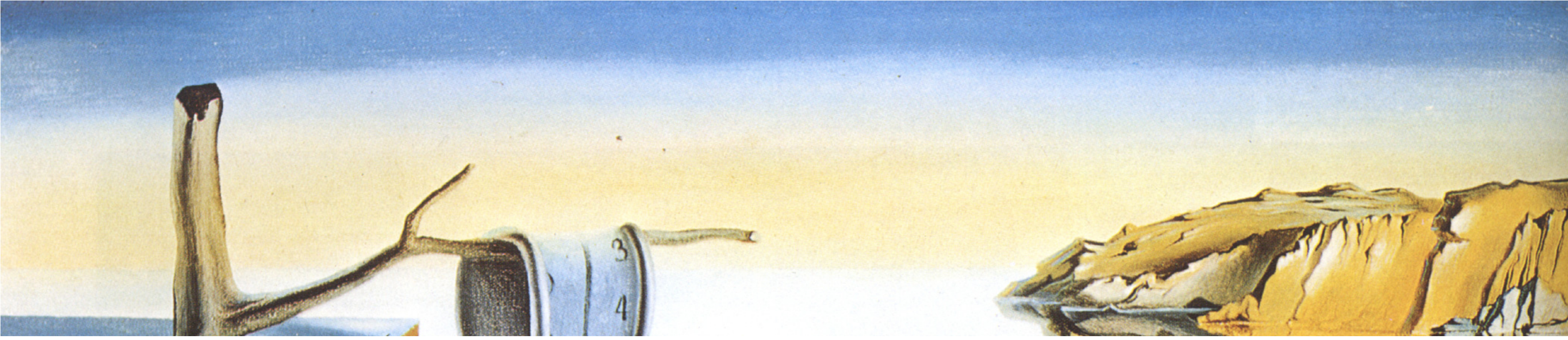
3.2.1 Concepts 7

4 Graphical User Interface 8



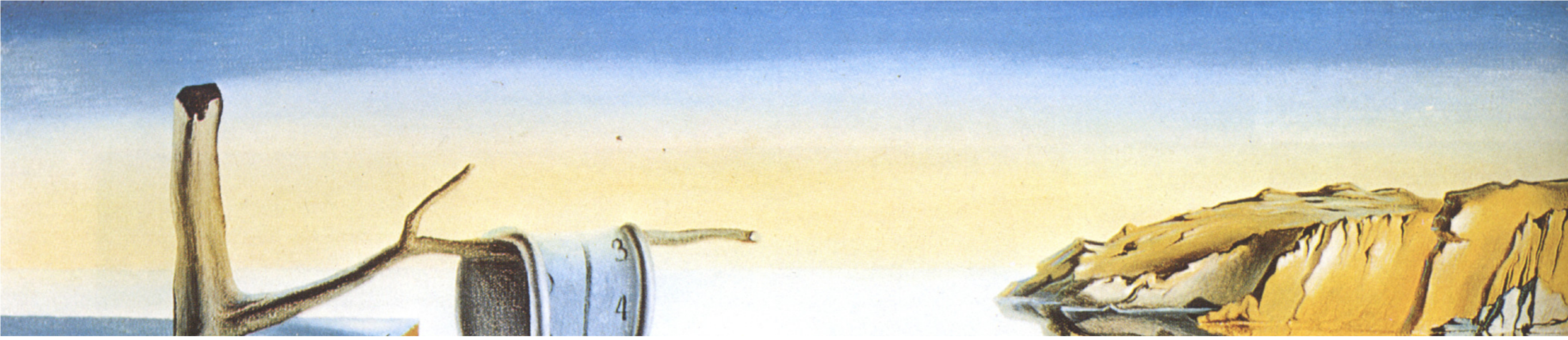
1. Introduction

1.1 Motivation



2. Structure

2.1 title



3. Power flow methods

3.1 Newton-Raphson-Iwamoto

3.2 Holomorphic Embedding (ASU)

First introduced by Antonio Trias in 2012 [1], promises to be a non-divergent power flow method. Trias originally developed a version with no voltage controlled nodes (PV), in which the convergence properties are excellent (With this software try to solve any grid without PV nodes to check this affirmation).

The version programmed in the file `HelmVect.py` has been adapted from the master thesis of Muthu Kumar Subramanian at the Arizona State University [2]. This version includes a formulation of the voltage controlled nodes. My experience indicates that the introduction of the PV control deteriorates the convergence properties of the holomorphic embedding method. However, in many cases, it is the best approximation to a solution. especially when Newton-Raphson does not provide one.

The `HelmVect.py` file contains a vectorized version of the algorithm. This means that the execution in python is much faster than a previous version that uses loops.

3.2.1 Concepts

All the power flow algorithms until the HELM method was introduced were iterative and recursive. The helm method is iterative but not recursive. A simple way to think of this is that traditional power flow methods are exploratory, while the HELM method is a planned journey. In theory the HELM method is superior, but in practice the numerical degeneration makes it less ideal.

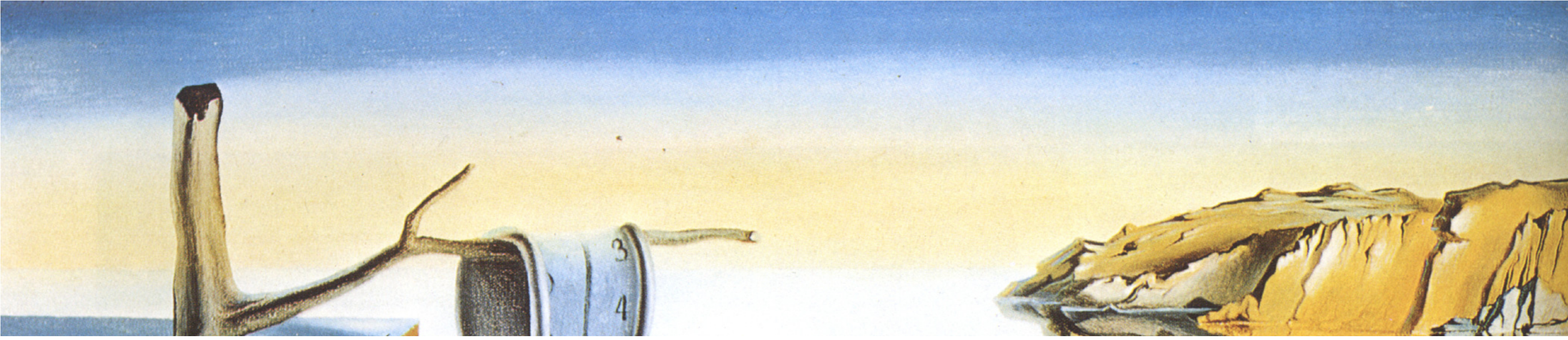
The fundamental idea of the recursive algorithms is that given a voltage initial point (1 p.u. at every node, usually) the algorithm explores the surroundings of the initial point until a suitable voltage solution is reached or no solution at all is found because the initial point is supposed to be "far" from the solution.

On the HELM methods, we form a "curve" that departures from a known mathematically exact solution that is obtained from solving the grid with no power injections. This is possible because with no power injections, the grid equations become linear and straight forward to solve. The arriving point of the "curve" is the solution that we want to achieve. That "curve" is best approximated by a Padè approximation. To compute the Padè approximation we need to compute coefficient of the unknown variables, in our case the voltage (and possibly the reactive power).

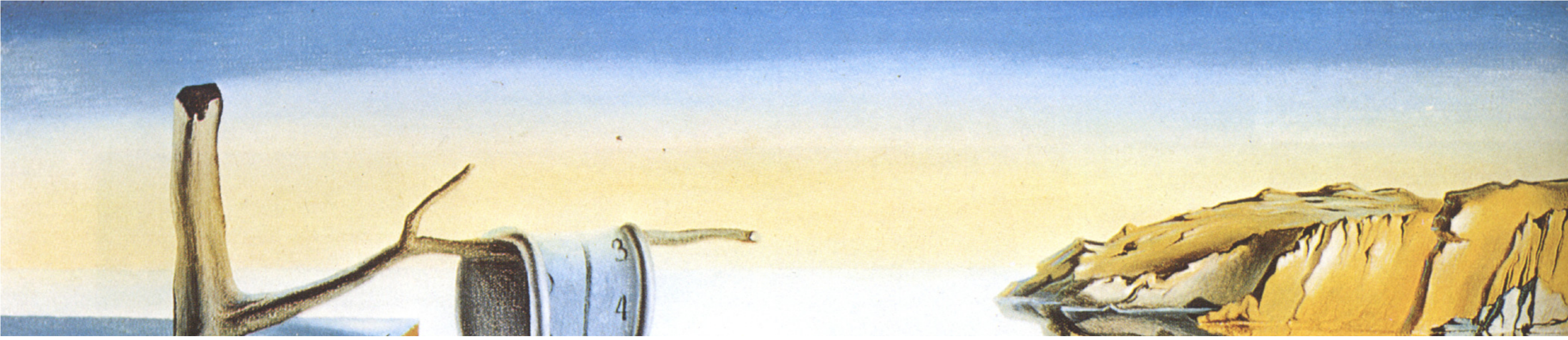
The HELM formulation consists in the derivation of formulas that enable the calculation of the coefficients of the series that describes the "curve" from the mathematically know solution to the unknown solution. Once the coefficients are obtained, the Padè approximation computes the voltage solution at the "end of the curve", providing the desired voltage solution. The more coefficients we compute the more exact the solution is (this is true until the numerical precision limit is reached).

All this sounds very strange, but it works ;)

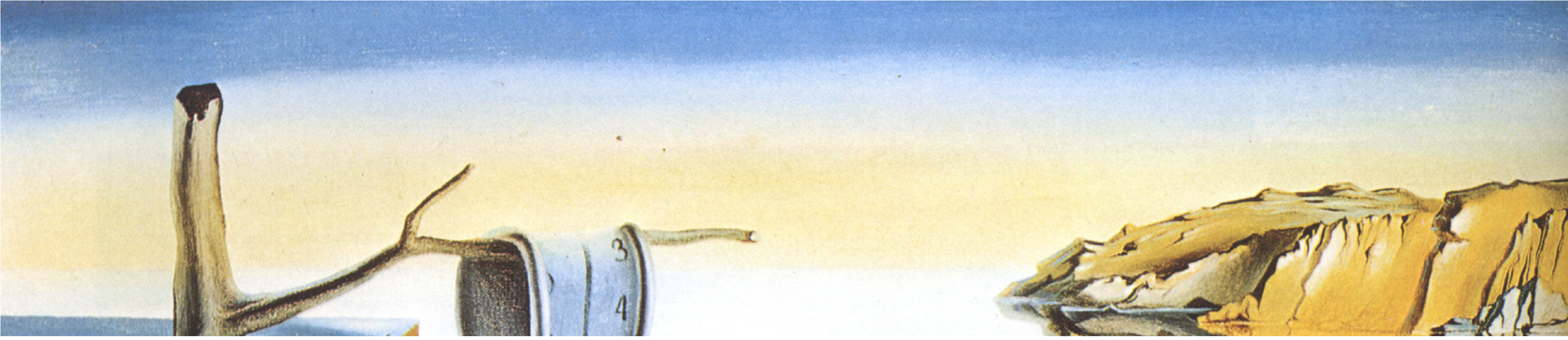
If you want to get familiar with this concept, you should read about the homotopy concept. In practice the continuation power flow does the same as the HELM algorithm, it takes a known solution and changes the loading factors until a solution for another state is reached.



4. Graphical User Interface



Bibliography



Bibliography

[1] A. Trias, “The holomorphic embedding load flow method,” pp. 1–8, July 2012.

[2] M. K. Subramanian, “Application of holomorphic embedding to the power-flow problem,” 2014.