Building *your own* fast solver

Homotopy Continuation Tutorial

Ricardo Fabbri **Rio de Janeiro State University**

Author of MiNuS github.com/rfabbri/minus

CVPR
JUNE 17-21, 2024
SEATTLE, WA

Building *your own* fast solver

MiNuS fast continuation solver framework

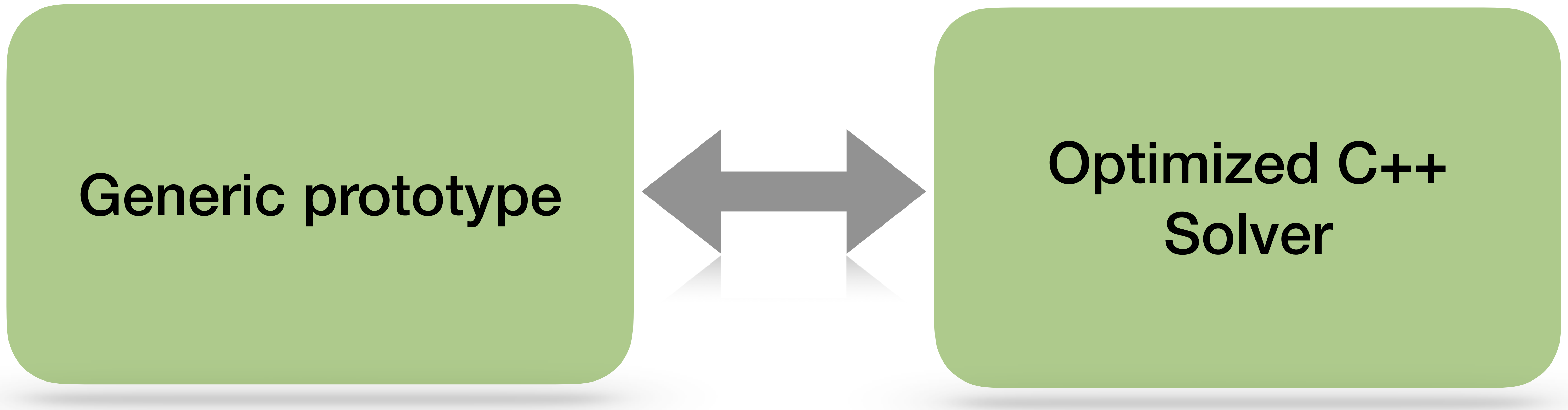
1. Build a Macaulay basic prototype

   macaulay2.com

2. Add Macaulay pro features
3. Build a C++ fast solver step-by-step

   github.com/rfabbri/minus

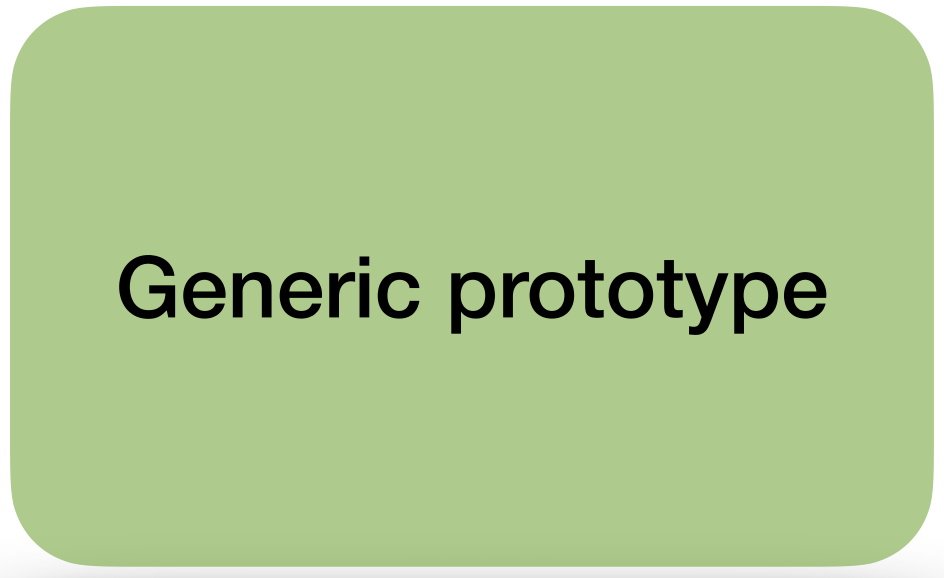
   1. Design of fast template system
   2. Design of fast executable
4. Benchmark



Building a Macaulay prototype

MiNuS fast continuation solver framework

- Macaulay is a powerful, modern scripting language
- Focuses on symbolic computation
- Runs Homotopy Continuation
    - Generic C++ under the hood
    - Code-matched with optimized MiNuS C++ framework
    - Origin of fastest most reliable solver for hard problems
- Used by key Algebraic-geometry researchers
    - Able to help build fast solvers that no one else has



macaulay2.com

github.com/rfabbri/minus

Building a C++ fast solver

MiNuS fast continuation solver framework

- MiNuS is a template system that helps build and debug your fast solver
- Allows to test diferent formulations, problems, and implementations
- Leverages the fact that the core algorithm is just predictor-corrector
- Born out of intense semester program at ICERM
- Bertini and Macaulay team on large unsolved problem
  - Solver with Macaulay 60s —> MiNuS in 400ms

Optimized C++
Solver

Trifocal Relative Pose from Lines at Points, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2022, *CVPR 2020 and PAMI 2022*

Ricardo Fabbri, Timothy Duff, Hongyi Fan, Margaret Regan, David de Pinho (my former MSc. Student), Elias Tsigaridas, Charles Wampler, Jonathan Hauenstein, Peter Giblin, Benjamin Kimia, Anton Leykin and Tomas Pajdla (pdf | code | datasets | bib)

github.com/rfabbri/minus

Building a C++ fast solver

MiNuS fast continuation solver framework

Main idea of MiNuS: Progressive specialization

- Key to good Numerical algorithms is knowledge to specialize algorithms
- HC is too generic
  - Start system, Randomization, Predictor-corrector, evaluator code are generic
- Pretraining Analogy
  - Existing numerical algorithms are pretrained on the space of all problems,
    - “Dataset” is very unbalanced towards offline problems
    - **Realtime scenarios underrepresented**
  - Fine-tuning the HC model to your problem is key

Optimized C++ Solver

github.com/rfabbri/minus

Building a C++ fast solver

MiNuS fast continuation solver framework

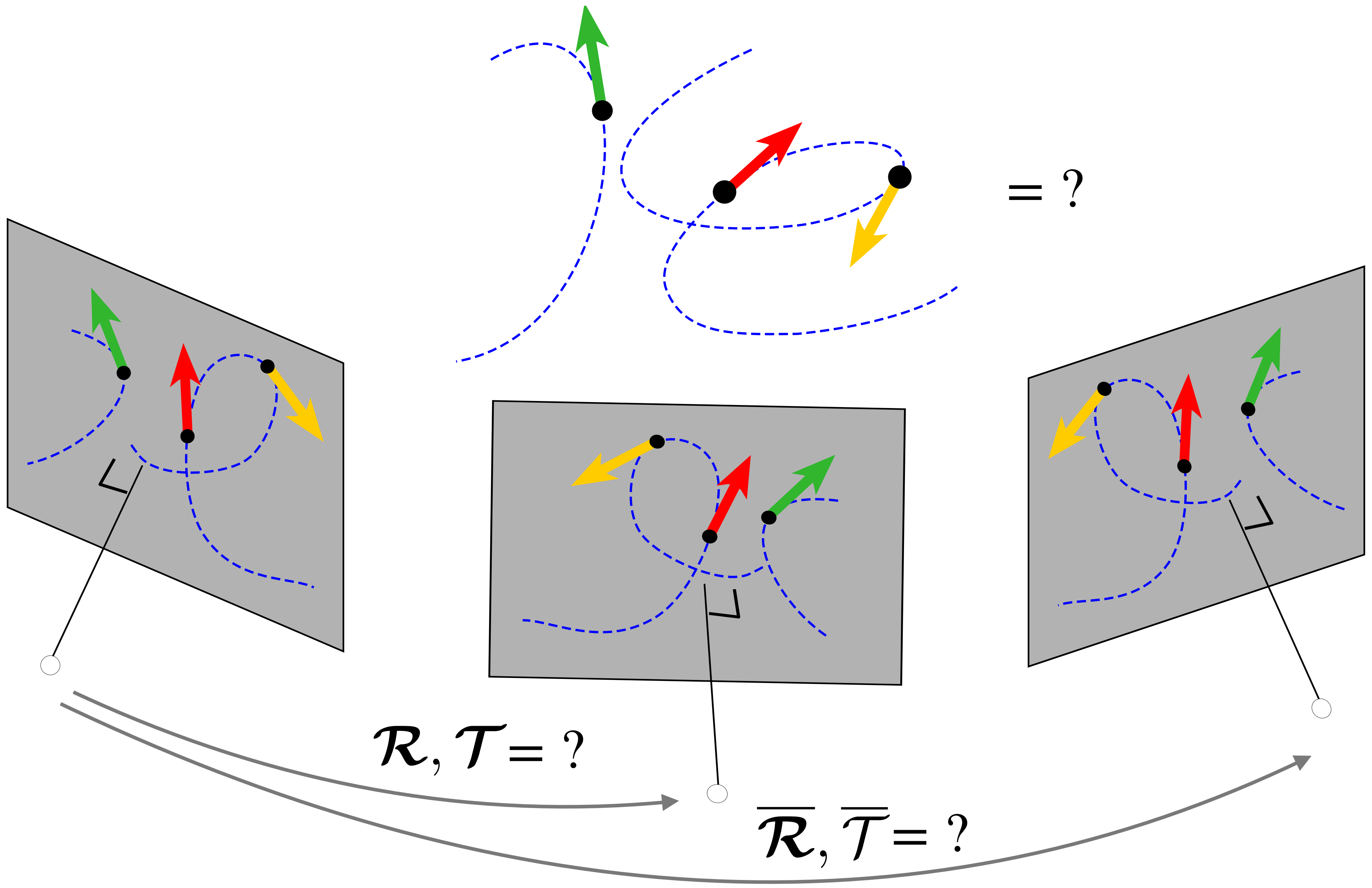
chicago14a

trifocal 14x14 formulation a

linecircle2a

line-circle 14x14 formulation a

Model problems

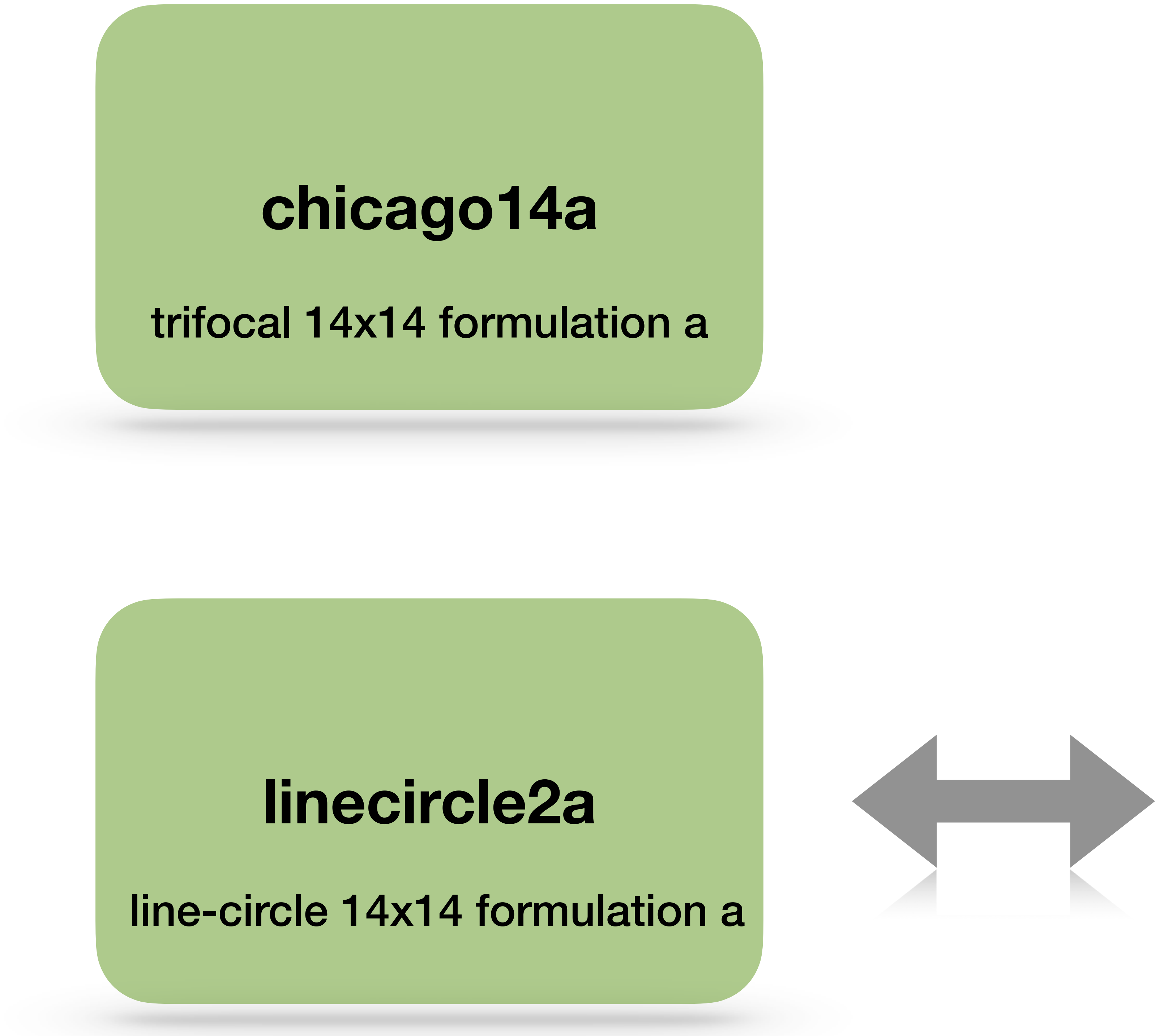

= ?
$\mathcal{R}, \mathcal{T} = ?$
$\overline{\mathcal{R}}, \overline{\mathcal{T}} = ?$

Trifocal relative pose from SIFT features
Proposed in Fabbri ECCV 2012
Solved by Fabbri, Duff, etal PAMI2022
Algebraic degree of nonlinearity 312

github.com/rfabbri/minus

Building a C++ fast solver

MiNuS fast continuation solver framework



chicago14a

trifocal 14x14 formulation a

linecircle2a

line-circle 14x14 formulation a

$$\begin{aligned}(x^2+y^2)+b*x+c&=0\\ d*x+e*y+f&=0\end{aligned}$$

Recal Tim Duff's talk

Model problems

github.com/rfabbri/minus

Building a Macaulay prototype

Hands-on

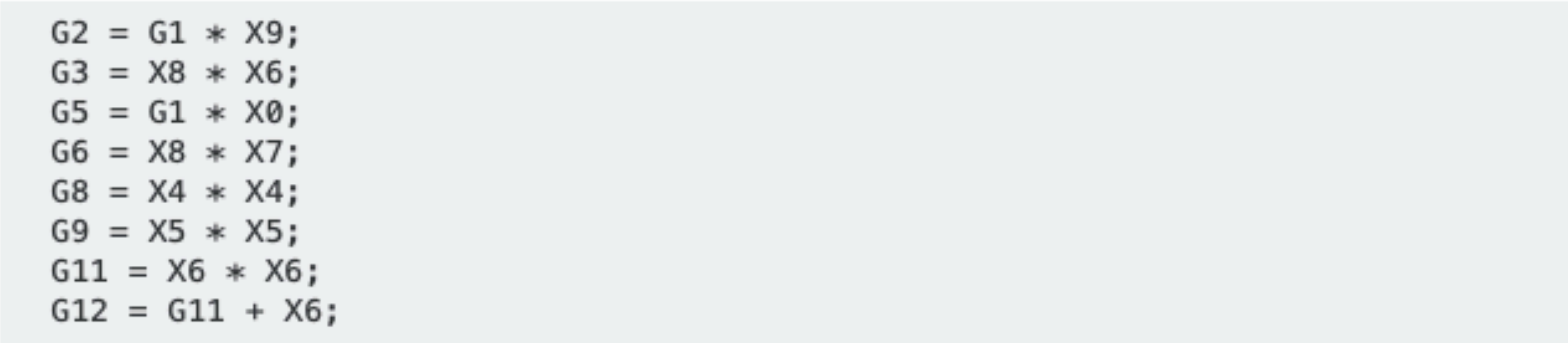
- git clone minus
- scripts are in minus/tutorial
- equations, start, end
- running
  - shell: m2
  - m2: load(“script”)

Generic prototype

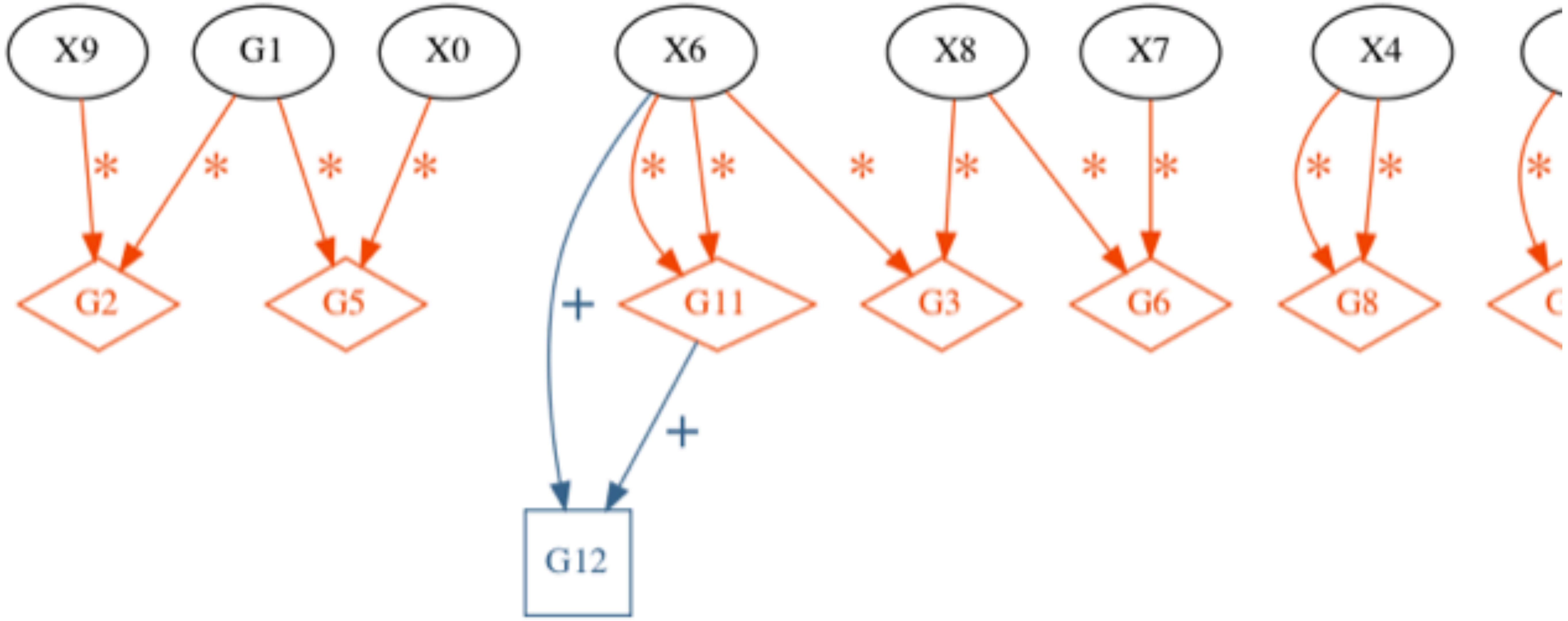
Building a Macaulay prototype

Hands-on

- Evaluators to C++



```
G2 = G1 * X9;
G3 = X8 * X6;
G5 = G1 * X0;
G6 = X8 * X7;
G8 = X4 * X4;
G9 = X5 * X5;
G11 = X6 * X6;
G12 = G11 + X6;
```

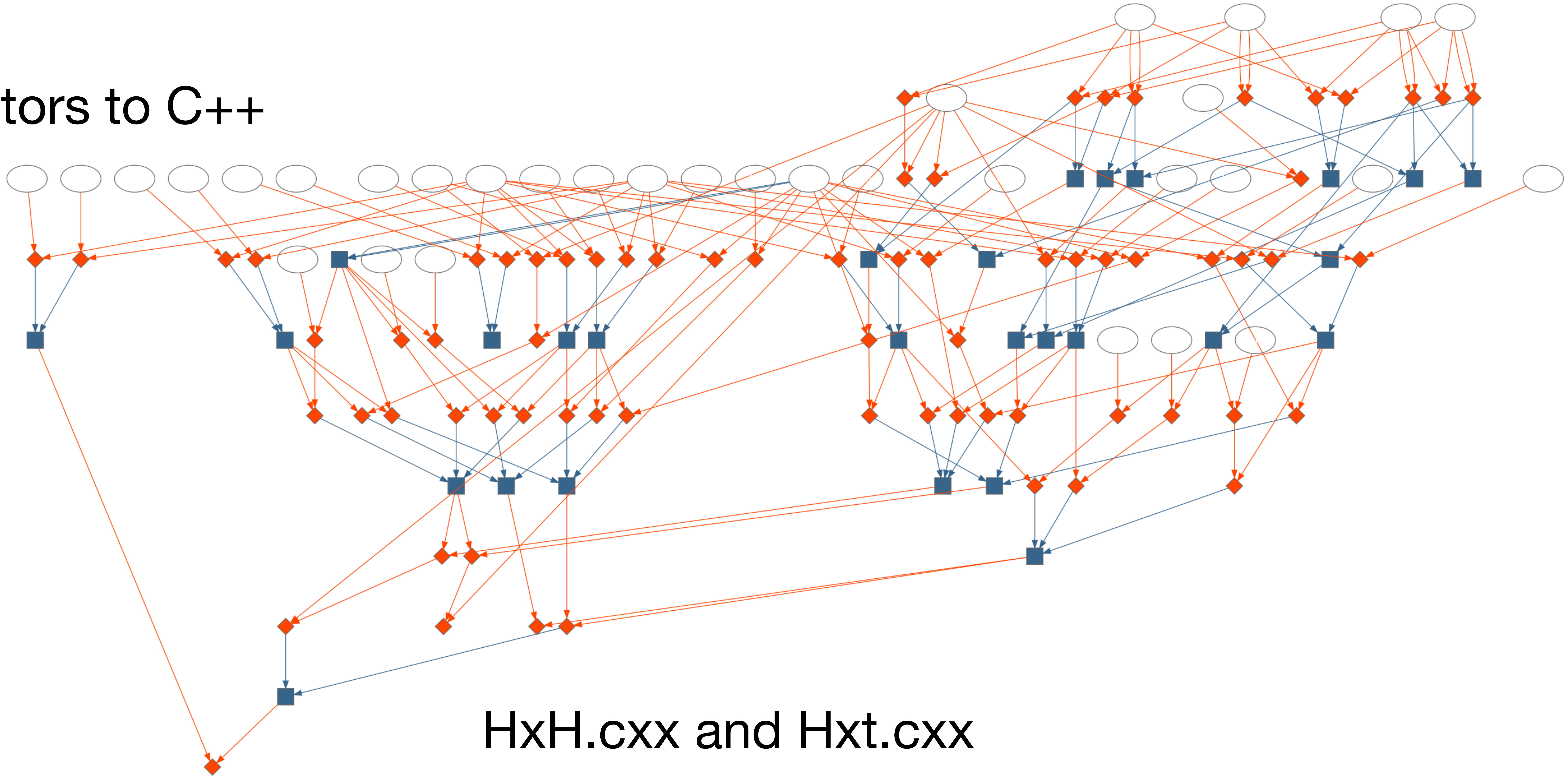


HxH.cxx and Hxt.cxx

Building a Macaulay prototype

Hands-on

- Evaluators to C++



HxH.cxx and Hxt.cxx

macaulay2.com

github.com/rfabbri/minus

— End or part 2 —