CONTROL COURSE 2025

Introduction to RAPTOR

Tutorial

Cassandre Contré *(cassandre.contre@epfl.ch)*

RAPTOR equations (recap)

current:

$$\sigma_{\parallel}\left(\frac{2\Phi_b\rho}{V'_\rho}\left.\frac{\partial\psi}{\partial t}\right|_\rho-\frac{\rho^2\dot{\Phi}_b}{V'_\rho}\frac{\partial\psi}{\partial\rho}\right)=\frac{F^2}{8\pi^2\mu_0\Phi_b V'_\rho}\frac{\partial}{\partial\rho}\left[\frac{g_2 g_3}{\rho}\frac{\partial\psi}{\partial\rho}\right]-\langle\mathbf{j_{ni}}\cdot\mathbf{B}\rangle$$

temperature:

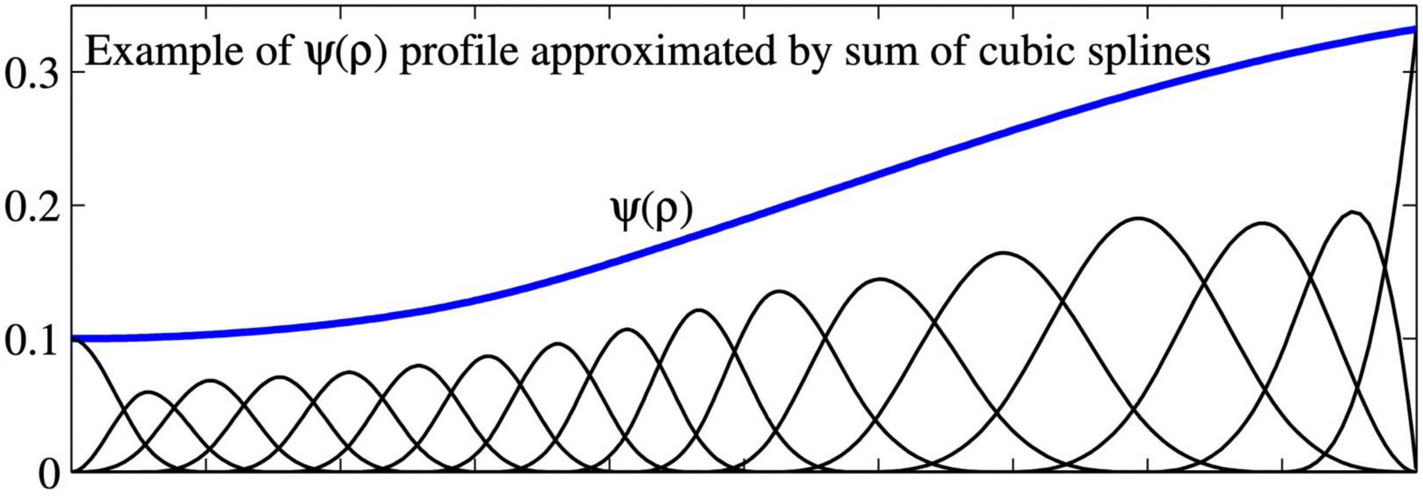
$$\frac{3}{2}(V'_\rho)^{-5/3}\left(\left.\frac{\partial}{\partial t}\right|_\rho-\frac{\dot{\Phi}_b}{2\Phi_b}\frac{\partial}{\partial\rho}\rho\right)[(V'_\rho)^{5/3}n_{e,i}T_{e,i}]+\frac{1}{V'_\rho}\frac{\partial}{\partial\rho}\left(-\frac{g_1}{V'_\rho}n_{e,i}\chi_{e,i}\frac{\partial T_{e,i}}{\partial\rho}+\frac{5}{2}T_{e,i}\Gamma_{e,i}g_0\right)=P_{e,i}$$

density:

$$\frac{1}{V'_{\hat{\rho}}}\left(\left.\frac{\partial}{\partial t}\right|_{\hat{\rho}}-\frac{\dot{\Phi}_b}{2\Phi_b}\frac{\partial}{\partial\hat{\rho}}\hat{\rho}\right)[V'_{\hat{\rho}}n_e]=-\frac{1}{V'_{\hat{\rho}}}\frac{\partial}{\partial\hat{\rho}}\Gamma_e+S_e$$

Spatial discretisation

$$\mathbf{x}(t) = \begin{bmatrix} \hat{\boldsymbol{\psi}}(t) \\ \hat{\mathbf{T}}_{\mathbf{e,i}}(t) \end{bmatrix}, \qquad \begin{bmatrix} \boldsymbol{\psi}(\rho, t) \\ \mathbf{T}_{\mathbf{e,i}}(\rho, \mathbf{t}) \end{bmatrix} = \sum_{\alpha=1}^{n_{sp}} \Lambda_\alpha(\rho) \begin{bmatrix} \hat{\boldsymbol{\psi}}(t) \\ \hat{\mathbf{T}}_{\mathbf{e,i}}(t) \end{bmatrix}$$



From continuous PDE to discrete ODE

For $y(t) = \{\psi(t), T_e(t), \dots\}$,

$$\frac{dy(\rho,t)}{dt} = \sum_{\alpha=1}^{n_{SP}} \Lambda_\alpha(\rho) \frac{d\hat{y}_\alpha(t)}{dt}$$

time derivative

$$\frac{\partial y(\rho,t)}{\partial \rho} = \sum_{\alpha=1}^{n_{SP}} \frac{\partial \Lambda_\alpha(\rho)}{\partial \rho} \hat{y}_\alpha(t)$$

space derivative

Coming back to the equations,

$$\sum_{\alpha=1}^{n_{SP}} m \frac{d\hat{y}_\alpha(t)}{dt} \Lambda_\alpha(\rho) = -\sum_{\alpha=1}^{n_{SP}} \hat{y}_\alpha(t) \frac{\partial}{\partial \rho} g \frac{\partial \Lambda_\alpha(\rho)}{\partial \rho} + kj$$

From continuous PDE to discrete ODE

For $y(t) = \{\psi(t), T_e(t), \dots\}$,

$$\frac{dy(\rho,t)}{dt} = \sum_{\alpha=1}^{n_{SP}} \Lambda_\alpha(\rho) \frac{d\hat{y}_\alpha(t)}{dt}$$

time derivative

$$\frac{\partial y(\rho,t)}{\partial \rho} = \sum_{\alpha=1}^{n_{SP}} \frac{\partial \Lambda_\alpha(\rho)}{\partial \rho} \hat{y}_\alpha(t)$$

space derivative

Coming back to the equations,

$$\sum_{\alpha=1}^{n_{SP}} m \frac{d\hat{y}_\alpha(t)}{dt} \Lambda_\beta(\rho) \Lambda_\alpha(\rho) = -\sum_{\alpha=1}^{n_{SP}} \hat{y}_\alpha(t)\, \Lambda_\beta(\rho) \frac{\partial}{\partial \rho} g \frac{\partial \Lambda_\alpha(\rho)}{\partial \rho} + kj$$

projection on $\Lambda_\beta(\rho)$

From continuous PDE to discrete ODE

For $y(t) = \{\psi(t), T_e(t), \dots\}$,

$$\frac{dy(\rho,t)}{dt} = \sum_{\alpha=1}^{n_{SP}} \Lambda_\alpha(\rho) \frac{d\hat{y}_\alpha(t)}{dt}$$

time derivative

$$\frac{\partial y(\rho,t)}{\partial \rho} = \sum_{\alpha=1}^{n_{SP}} \frac{\partial \Lambda_\alpha(\rho)}{\partial \rho} \hat{y}_\alpha(t)$$

space derivative

Coming back to the equations,

$$\sum_{\alpha=1}^{n_{SP}} m \frac{d\hat{y}_\alpha(t)}{dt} \int_0^{\rho_e} \Lambda_\beta(\rho)\Lambda_\alpha(\rho) d\rho = -\sum_{\alpha=1}^{n_{SP}} \hat{y}_\alpha(t) \int_0^{\rho_e} \Lambda_\beta(\rho) \frac{\partial}{\partial \rho} g \frac{\partial \Lambda_\alpha(\rho)}{\partial \rho} d\rho + \int_0^{\rho_e} k\, j d\rho$$

integral over ρ

From continuous PDE to discrete ODE

For $y(t) = \{\psi(t), T_e(t), \dots\}$,

$$\frac{dy(\rho,t)}{dt} = \sum_{\alpha=1}^{n_{SP}} \Lambda_\alpha(\rho) \frac{d\hat{y}_\alpha(t)}{dt}$$

time derivative

$$\frac{\partial y(\rho,t)}{\partial \rho} = \sum_{\alpha=1}^{n_{SP}} \frac{\partial \Lambda_\alpha(\rho)}{\partial \rho} \hat{y}_\alpha(t)$$

space derivative

Coming back to the equations,

$$\sum_{\alpha=1}^{n_{SP}} m \frac{d\hat{y}_\alpha(t)}{dt} \int_0^{\rho_e} \Lambda_\beta(\rho)\Lambda_\alpha(\rho) d\rho = -\sum_{\alpha=1}^{n_{SP}} \hat{y}_\alpha(t) \left(\int_0^{\rho_e} g \frac{\partial \Lambda_\beta(\rho)}{\partial \rho} \frac{\partial \Lambda_\alpha(\rho)}{\partial \rho} d\rho + \left[g \Lambda_\beta(\rho) \frac{\partial \Lambda_\alpha(\rho)}{\partial \rho} \right]_0^{\rho_e} \right) + \int_0^{\rho_e} k\, j d\rho$$

integration by parts

From continuous PDE to discrete ODE

For $y(t) = \{\psi(t), T_e(t), \dots\}$,

$$\frac{dy(\rho,t)}{dt} = \sum_{\alpha=1}^{n_{SP}} \Lambda_\alpha(\rho) \frac{d\hat{y}_\alpha(t)}{dt}$$

time derivative

$$\frac{\partial y(\rho,t)}{\partial \rho} = \sum_{\alpha=1}^{n_{SP}} \frac{\partial \Lambda_\alpha(\rho)}{\partial \rho} \hat{y}_\alpha(t)$$

space derivative

Coming back to the equations,

$$\sum_{\alpha=1}^{n_{SP}} m \frac{d\hat{y}_\alpha(t)}{dt} \int_0^{\rho_e} \Lambda_\beta(\rho)\Lambda_\alpha(\rho) d\rho = -\sum_{\alpha=1}^{n_{SP}} \hat{y}_\alpha(t) \left(\int_0^{\rho_e} g \frac{\partial \Lambda_\beta(\rho)}{\partial \rho} \frac{\partial \Lambda_\alpha(\rho)}{\partial \rho} d\rho + \left[g \Lambda_\beta(\rho) \frac{\partial \Lambda_\alpha(\rho)}{\partial \rho} \right]_0^{\rho_e} \right) + \int_0^{\rho_e} k\, j d\rho$$

Matrix-vector ODE

From continuous PDE to discrete ODE

For $y(t) = \{\psi(t), T_e(t), \dots\}$,

$$\frac{dy(\rho,t)}{dt} = \sum_{\alpha=1}^{n_{SP}} \Lambda_\alpha(\rho) \frac{d\hat{y}_\alpha(t)}{dt}$$

time derivative

$$\frac{\partial y(\rho,t)}{\partial \rho} = \sum_{\alpha=1}^{n_{SP}} \frac{\partial \Lambda_\alpha(\rho)}{\partial \rho} \hat{y}_\alpha(t)$$

space derivative

Coming back to the equations,

$$\sum_{\alpha=1}^{n_{SP}} m \frac{d\hat{y}_\alpha(t)}{dt} \int_0^{\rho_e} \Lambda_\beta(\rho)\Lambda_\alpha(\rho) d\rho = -\sum_{\alpha=1}^{n_{SP}} \hat{y}_\alpha(t) \left(\int_0^{\rho_e} g \frac{\partial \Lambda_\beta(\rho)}{\partial \rho} \frac{\partial \Lambda_\alpha(\rho)}{\partial \rho} d\rho + \left[g \Lambda_\beta(\rho) \frac{\partial \Lambda_\alpha(\rho)}{\partial \rho} \right]_0^{\rho_e} \right) + \int_0^{\rho_e} k\, j d\rho$$

$$\bigtriangledown$$

$$f(\dot{x}(t), x(t), u(t)) = 0 \;\forall\; t$$

state evolution equation

Implicit scheme

In the interval $[t_k \; t_{k+1}]$, where do you evaluate your time derivative?

Forward Euler method (explicit)

$$\dot{y}_k \approx \frac{y_{k+1} + y_k}{\Delta t}, \qquad y_{k+1} = y_k + \Delta\mathrm{t}\ \dot{y}_k$$

Backward Euler Method (implicit)

$$\dot{y}_{k+1} \approx \frac{y_{k+1} + y_k}{\Delta t}, \qquad y_{k+1} = y_k + \Delta\mathrm{t}\ \dot{y}_{k+1}$$

Semi-implicit Euler Method

$$\alpha\dot{y}_{k+1} + (1-\alpha)\dot{y}_k \approx \frac{y_{k+1} + y_k}{\Delta t}, \qquad y_{k+1} = y_k + \Delta\mathrm{t}\ (\alpha\dot{y}_{k+1} + (1-\alpha)\dot{y}_k)$$

Gradient descent

How do you evolve your state $x(t)$?

given $u_k \ \forall\ k$, initial state x_0 and initial state sensitivity $\partial x_0/\partial p$.
for each time step $k = 0 \dots M$ **do**
  First guess $x_{k+1}^{(0)} = x_k + (x_k - x_{k-1})$; $i = 0$
  repeat
    Evaluate $\tilde{f}_k^{(i)}(x_{k+1}^{(i)}, x_k, u_k)$ (Function residual)
    Evaluate $\mathcal{J}_{k+1} = \partial \tilde{f}_k/\partial x_{k+1}$ (Jacobian)
    Solve Eq. (D.34) for d (Newton direction) $\longrightarrow$ $\mathcal{J}_{k+1}^k d = \tilde{f}_k,$
    Update $x_{k+1}^{(i+1)} = x_{k+1}^{(i)} + \tau d$ (Newton step)
    $i \leftarrow i + 1$
  until $\tilde{f}^{(i)} < \epsilon_{Newton}$
  Set $x_{k+1} = x_{k+1}^{(i)}$
  Evaluate $\mathcal{J}_k = \partial \tilde{f}_k/\partial x_k$
  Solve Eq.(7.46) for $\partial x_{k+1}/\partial p$ (Forward sensitivity)
end for

Parameter sensitivity & local linearization

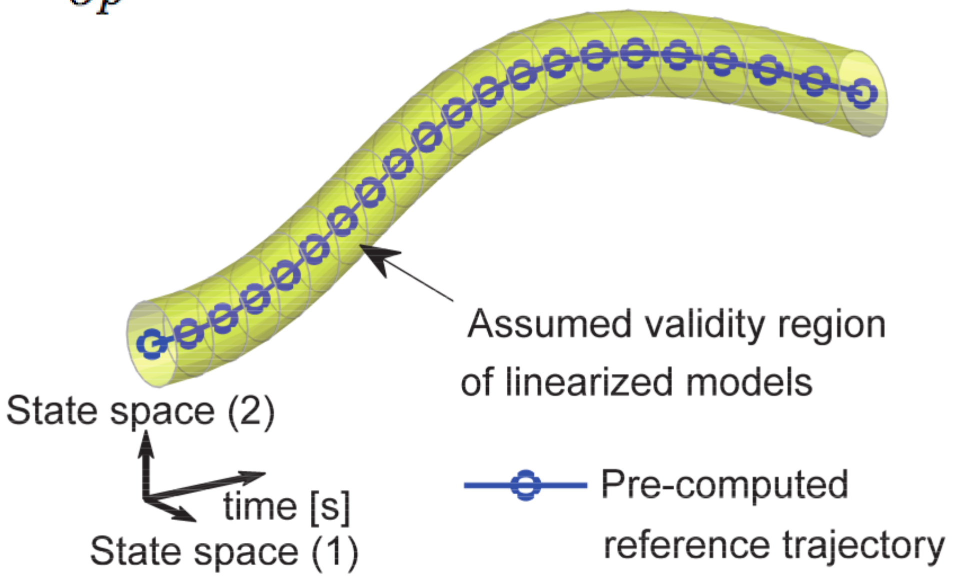
Sensitivity equation

$$0 = \frac{\mathrm{d}\tilde{f}_k}{\mathrm{d}p} = \boxed{\frac{\partial \tilde{f}_k}{\partial x_{k+1}}} \frac{\partial x_{k+1}}{\partial p} + \frac{\partial \tilde{f}_k}{\partial x_k}\frac{\partial x_k}{\partial p} + \frac{\partial \tilde{f}_k}{\partial u_k}\frac{\partial u_k}{\partial p} + \frac{\partial \tilde{f}_k}{\partial p}$$

known from Newton step

State linearisation

$$T_e(\rho,t)|_{p=p_0+\delta p} \approx T_e(\rho,t)_{p_0} + \frac{\partial T_e}{\partial x}\frac{\partial x}{\partial p}\delta p$$



Git repository

What's inside…

RAPTOR[1]/

[1] https://gitlab.epfl.ch/spc/raptor/raptor/-/tree/main

Git repository

What's inside…

RAPTOR[1]/

- `code`: core files
- `tests`: tests to check that code runs correctly
- `demos`: demonstration files and tutorials
- `doc`: documentation
- `equils`: equilibrium files defining equilibrium profiles
- `projects`: projects that are built "on top of" RAPTOR
- `optimization`: tools for non-linear optimization
- `data`: store temporary data here (not versioned)
- `personal`: store personal scripts here (not versioned)
- `license`: license agreement
- `RT`: Real-time implementations (Simulink)

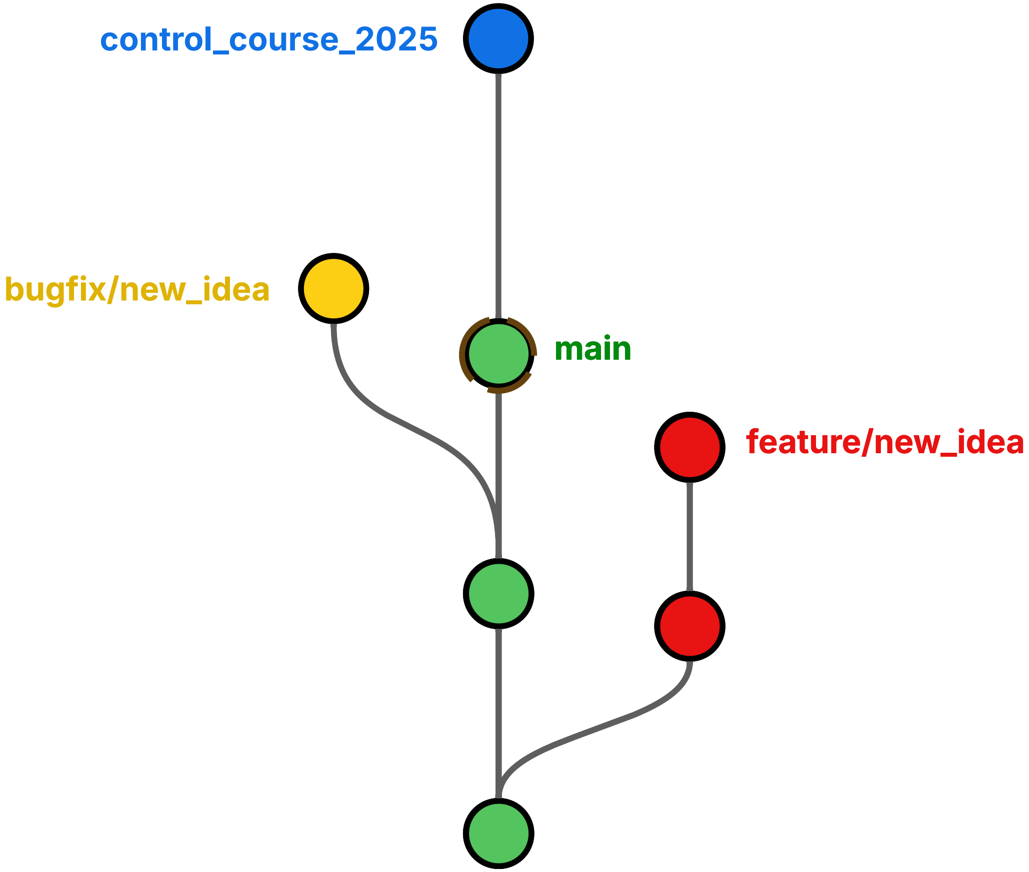
→ open source!

[1] https://gitlab.epfl.ch/spc/raptor/raptor/-/tree/main

Git repository

Gitlab

→ version control
→ collaborative interface
→ code testing



How to get started?

- many tutorials available on internet
- funny game: learngitbranching.js.org

Git repository

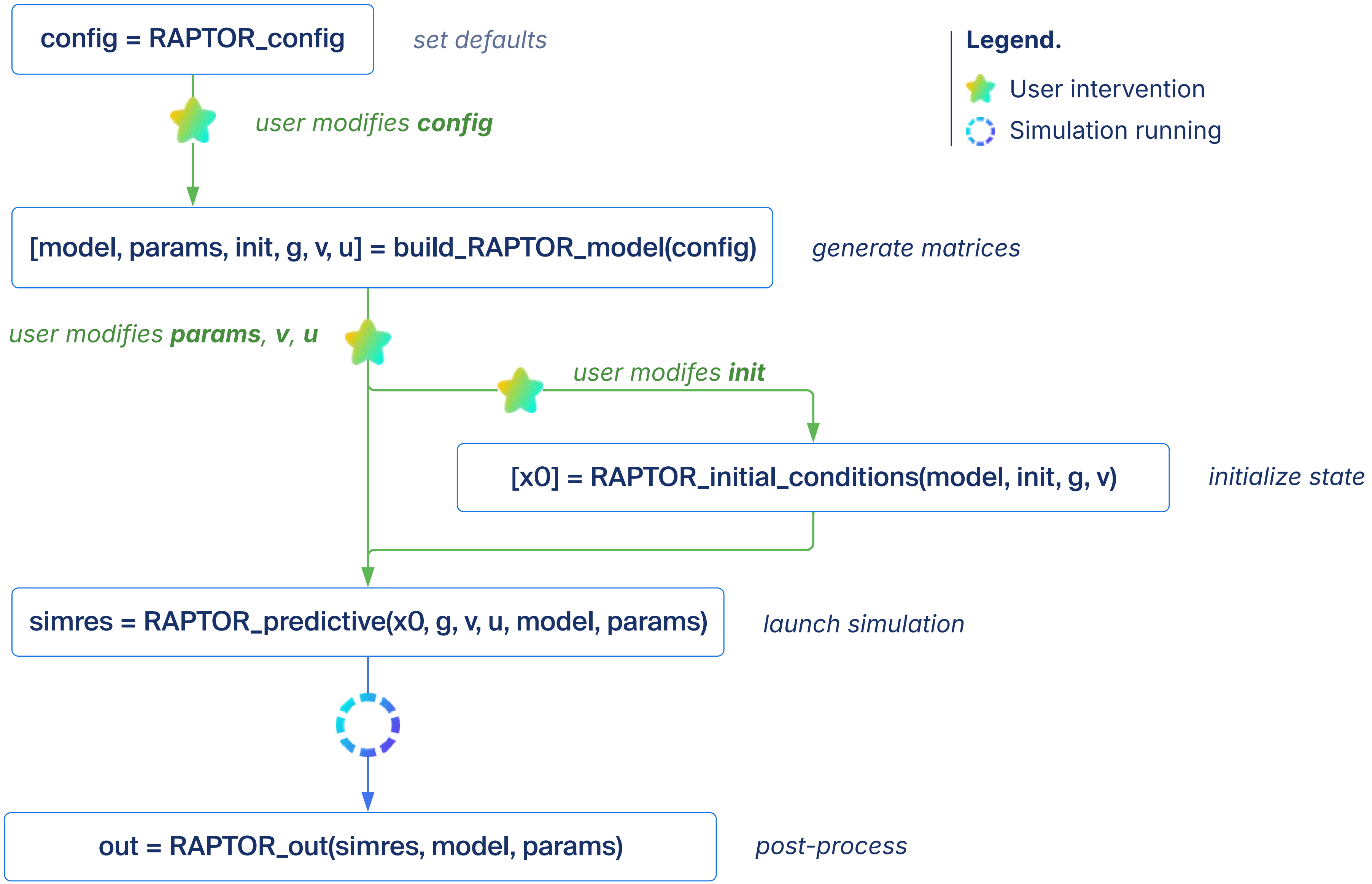
What's inside…

RAPTOR[1]/

- `code`: core files
- `tests`: tests to check that code runs correctly
- `demos`: demonstration files and tutorials where you will play today!
- `doc`: documentation
- `equils`: equilibrium files defining equilibrium profiles
- `projects`: projects that are built "on top of" RAPTOR
- `optimization`: tools for non-linear optimization
- `data`: store temporary data here (not versioned)
- `personal`: store personal scripts here (not versioned)
- `license`: license agreement
- `RT`: Real-time implementations (Simulink)

[1] https://gitlab.epfl.ch/spc/raptor/raptor/-/tree/main

Code structure



Code structure

config — specify equations to solve, equilibrium data, numerics

[model, params, init, g, v, u]

pre-calculated structure & matrices, <u>not to be changed manually</u>

tunable parameters

parameters for generating initial state x0

geometries

pre-known (time-varying) kinetic profile quantities

actuator inputs

Getting started 💪

- Tutorials and demo files
  - in `/demos`, execute `echodemo` or open the Matlab GUI to run the tutorials
  - You also can play with some standard scenarios in `/demos/standard_scenarios`
- Meaning of parameters can be found in functions comments
  - `RAPTOR_config` for general parameters
  - Inside the module to which the parameter applies

Documentation

- Physics and numerics
  - [Felici PPCF 2012, Felici PhD thesis, Felici NF 2018]
- Equation details
  - RAPTOR equation document in /doc directory

à vous de jouer !