TRAVEL-TIME SENSITIVITY KERNELS AND CALCULATING THEM ON THE WEB

Yacine Saidi

Max-Planck-Institut für Sonnensystemforschung, Katlenburg-Lindau, Germany

with contributions from Aaron Birch, Laurent Gizon, Jason Jackiewicz, Marcus Roth.

Max Planck Institute for
Solar System Research

Time-Distance Helioseismology

Travel times:

$$\delta\tau(\vec{x}_1,\vec{x}_2)=\sum_{\alpha}\int_S \delta q_\alpha(\vec{x})K^{\alpha}(\vec{x}_1,\vec{x}_2;\vec{x})d^3\vec{x}$$

Use forward modeling to find the sensitivity kernel *K*.

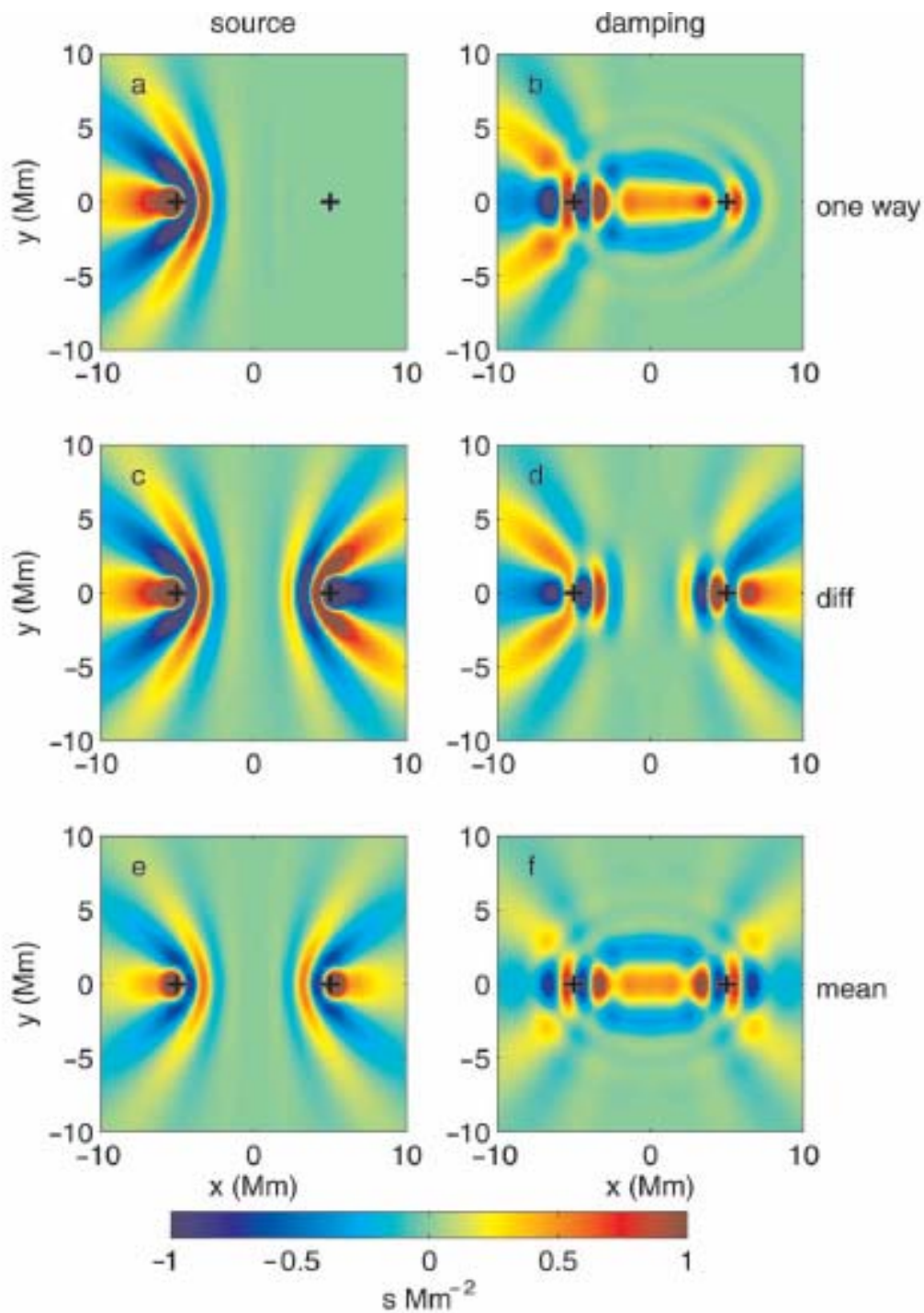
This is not that easy, unless one wants ray kernels.

Inversion procedures are well understood and require the use of, in particular, complicated 2-D and 3-D kernels (for high resolution inversions). We have a general procedure for calculating travel-time sensitivity kernels in the Born approximation (Gizon & Birch 2002), as well as other types of kernels.

There is a (growing) list of different types of sensitivity kernels: Damping and source amplitude, sound speed, 3-D flows, 2-D flows, 3-D flows in spherical geometry ...

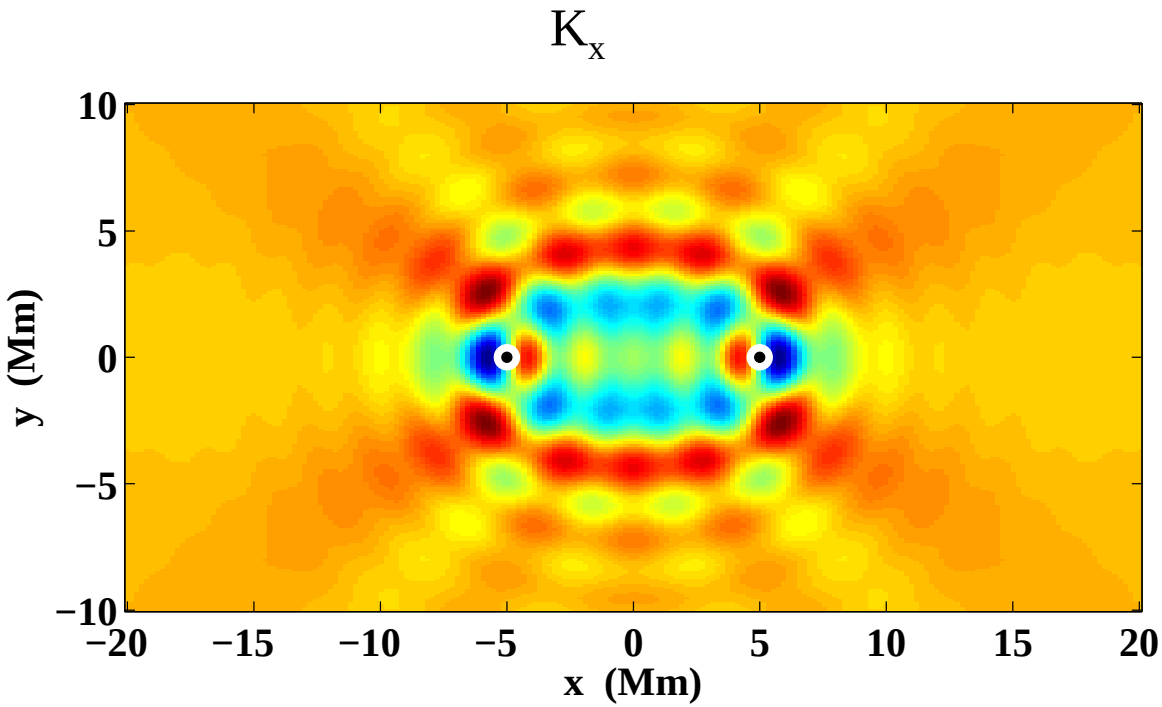
However, it is quite computationally expensive to compute most of these types of kernels, and also difficult to code.

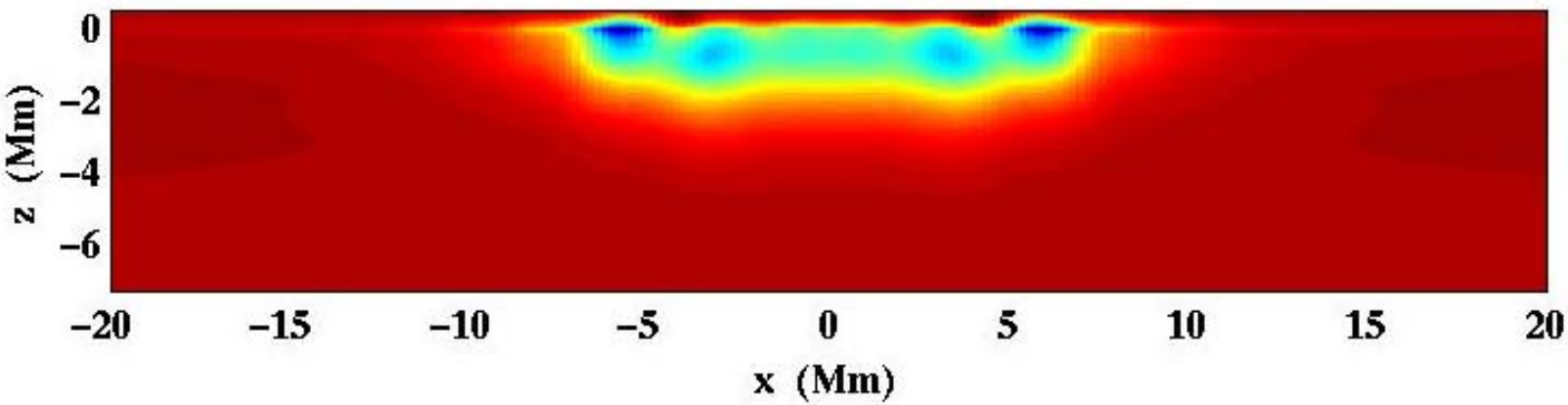
Source and damping strength sensitivity kernels using f modes.



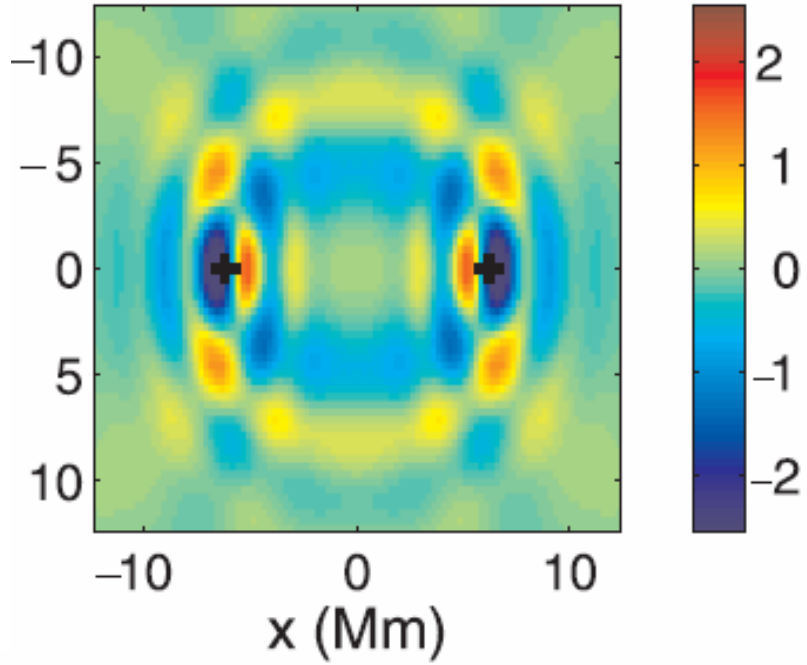
Gizon & Birch, ApJ. **571**, 966 (2002).

3-D sensitivity kernels for flows using f and p modes.

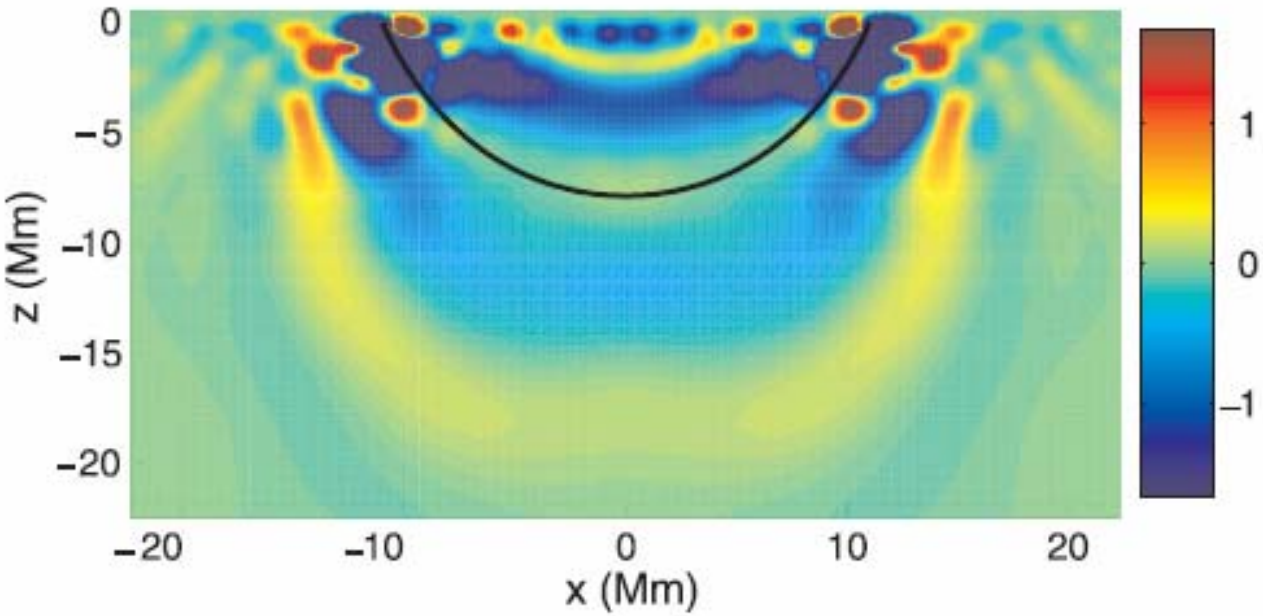




3-D kernels for sound speed using p modes.



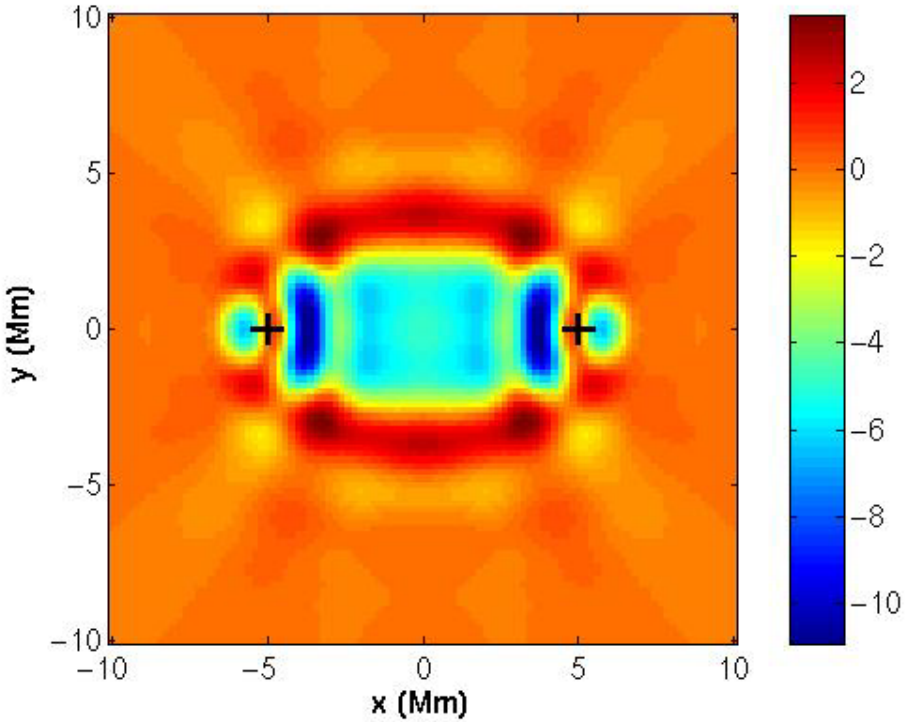
surface



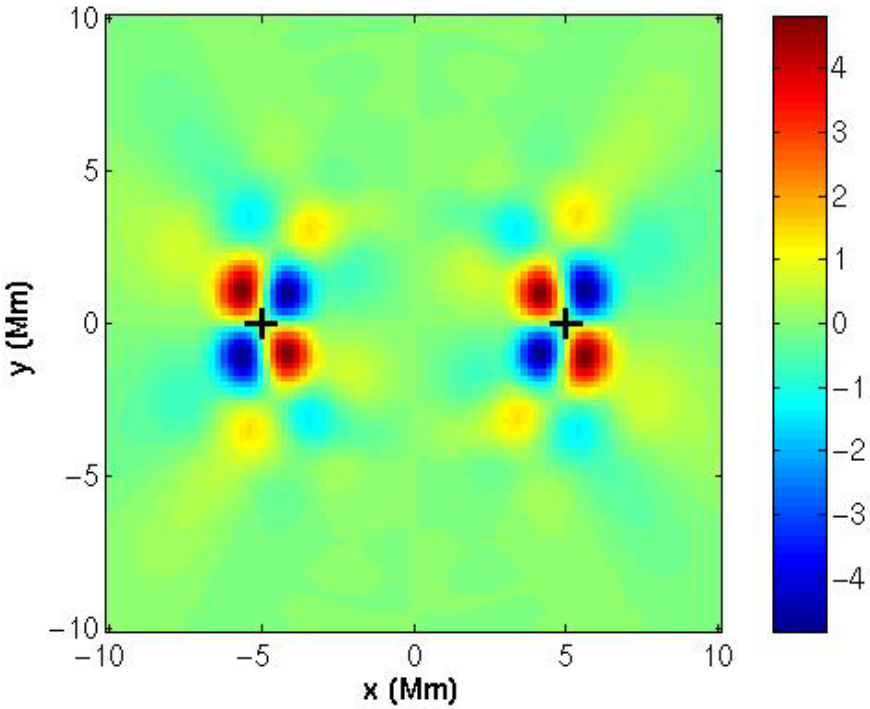
Birch et al., ApJ. **608**, 580 (2005).

2-D sensitivity kernels for horizontal flows using f modes.

K_x

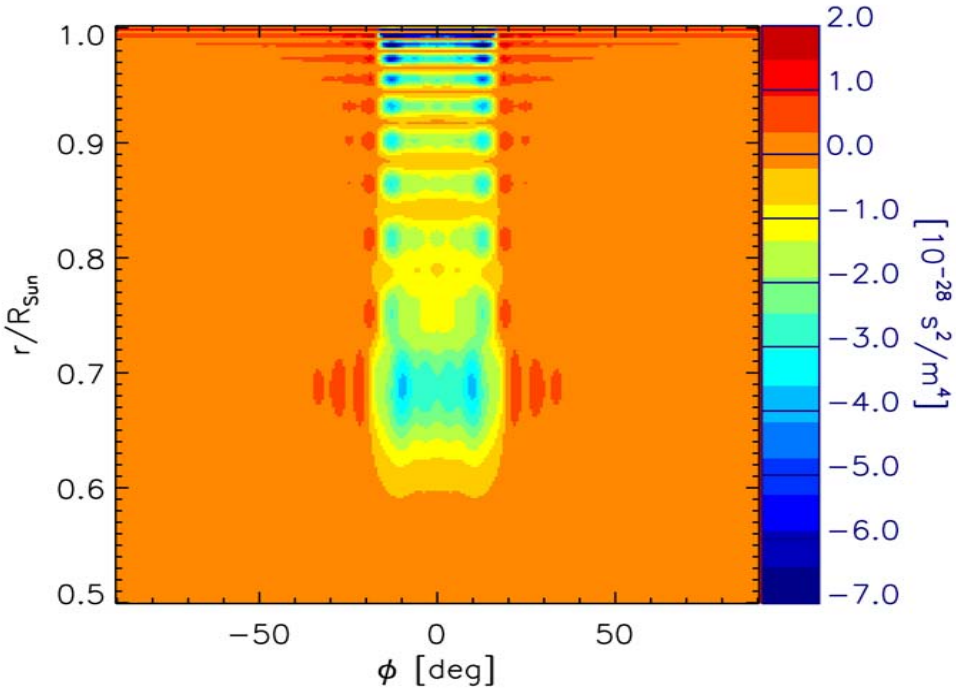


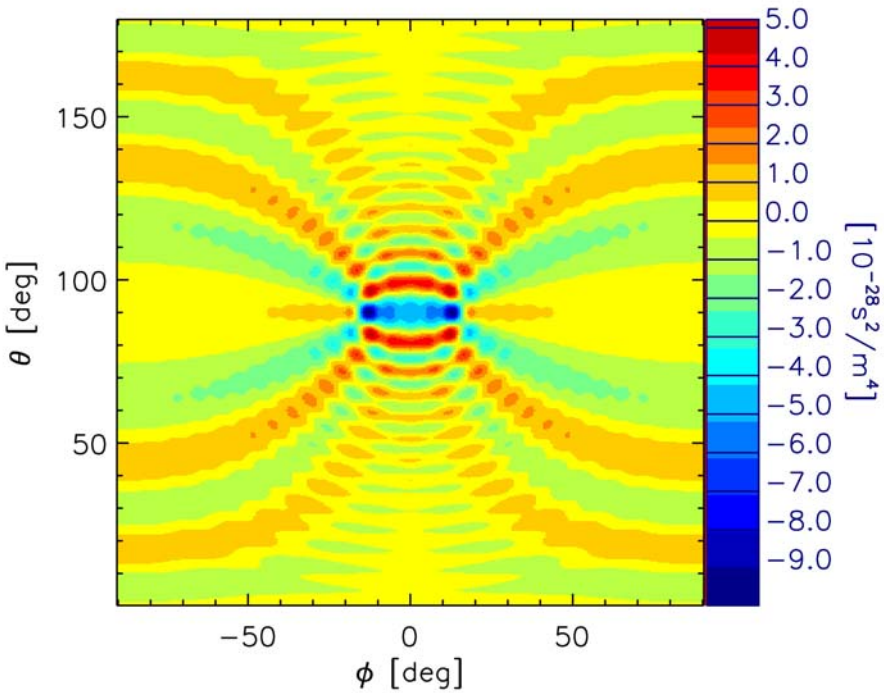
K_y



Jackiewicz, Gizon, Birch, ESA SP-617 (2006).

3-D kernels for flows in spherical geometry using p modes.





Roth & Gizon, ESA SP-624 (submitted).

Our goal for HELAS is to allow users to carry out the computation of the various kernels over the web.

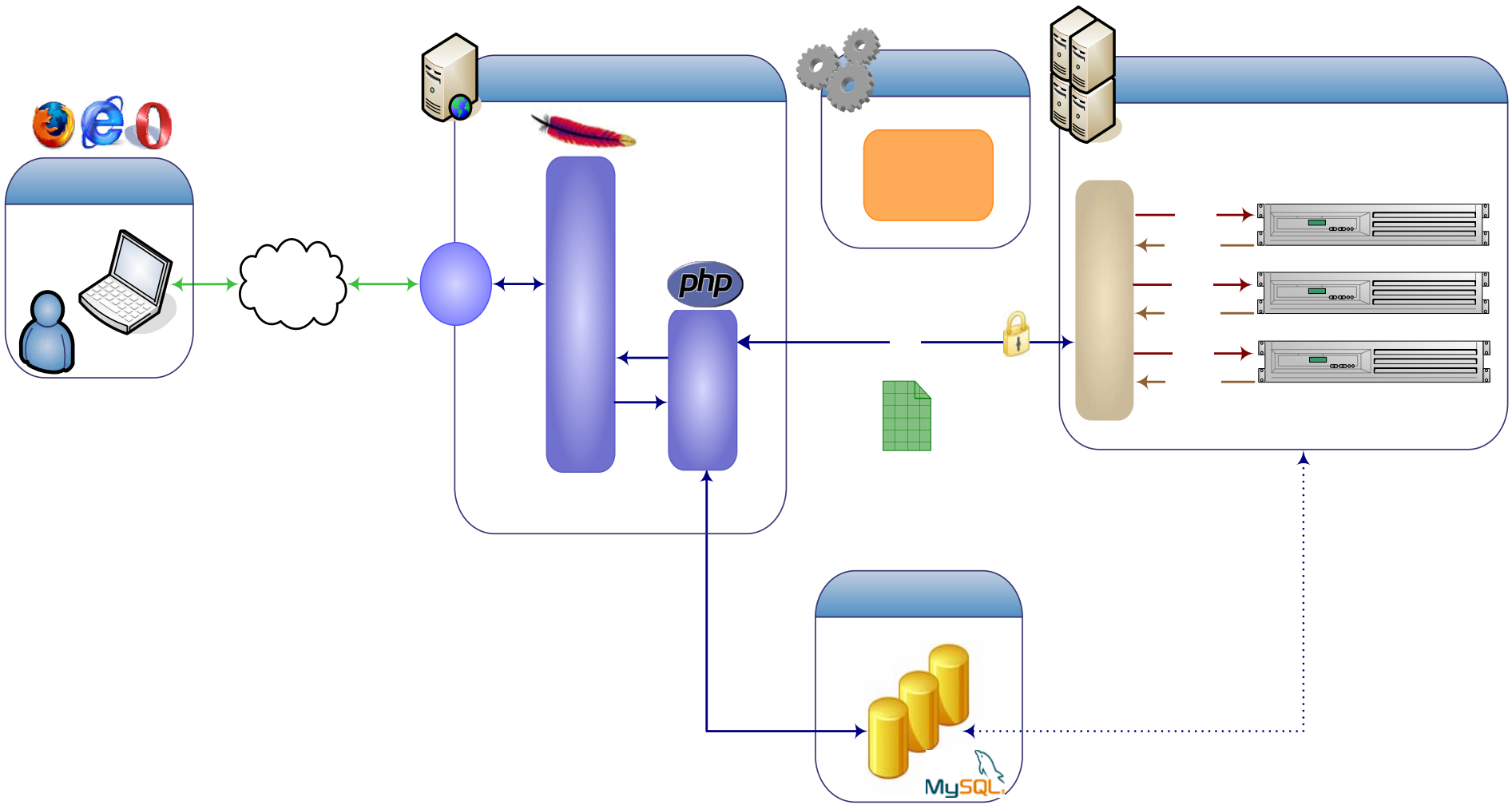
Some issues:

- We have codes for the kernels in **c++, IDL**, and **Matlab**.

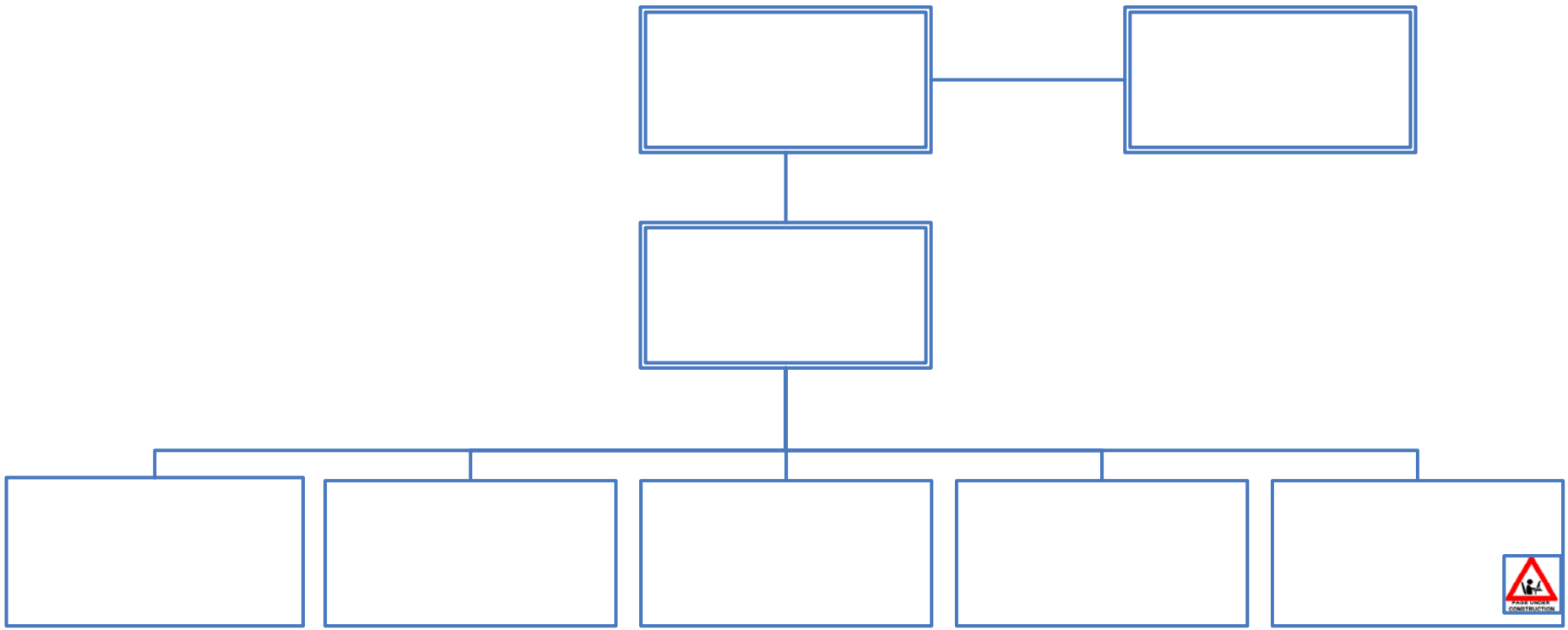
- We need to accommodate different codes written in different languages.

- Optimization: queuing, parallelization?

Layout of the computational components



Schematic of the user’s options



Demonstration

http://www.mps.mpg.de/services/seismo/

More Work (for the kernel facility):

- Database (kernels, Green's functions)

- User inputs: Filters (MTF, phase speed), background model, source, damping ...

- Documentation