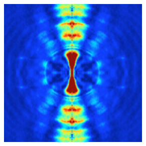
ISMB 2021

Fast high-resolution method for forward and inverse Abel transforms of velocity-map images

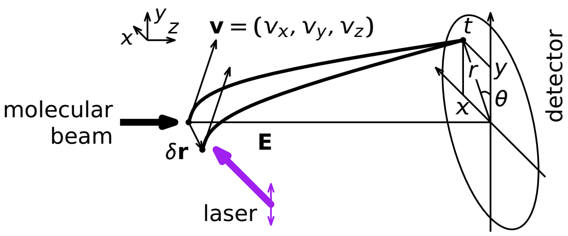
Mikhail Ryazanov

JILA, 440 UCB, Boulder, CO 80309, USA



PyAbel at GitHub

Velocity-map imaging



$\mathbf{v} \mapsto (x, y, t)$ [Chandler, Houston (1987)]
independent of $\delta\mathbf{r}$ [Eppink, Parker (1997)]

$$\begin{cases}(x, y) \approx \alpha_{\text{rad}} \cdot t_0 \cdot (v_x, v_y) \\ t \approx t_0 - \alpha_{\text{ax}} \cdot v_z\end{cases} \qquad \begin{cases}r \sim v_{\text{rad}} \\ t \sim v_{\text{ax}} + \text{const}\end{cases}$$

Cylindrical symmetry: $I(\mathbf{v}) = I(v, \theta)$
(axis ∥ polarization) $I(\theta) \sim P_n(\cos\theta)$

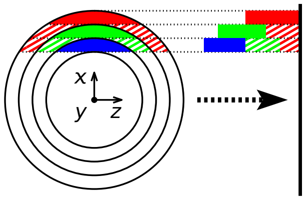
Abel transform

Cylindrically symmetric distribution

$$I(x, y, z) = I(\rho, y) \qquad \rho = \sqrt{x^2 + z^2}, \quad x = \rho\cos\varphi, \quad z = \rho\sin\varphi$$

Forward transform is a parallel projection:

$$P(x, y) = \int_{-\infty}^{+\infty} I(x, y, z)\,dz = \int_{|x|}^{\infty} I(\rho, y)\frac{\rho\,d\rho}{\sqrt{\rho^2 - x^2}}$$



Inverse transform is formally

$$I(\rho, y) = -\frac{1}{\pi}\int_{\rho}^{\infty} \frac{dP(x,y)}{dx}\frac{1}{\sqrt{x^2 - \rho^2}}\,dx$$

In practice: noisy singularity

- explicit basis set
- regularization

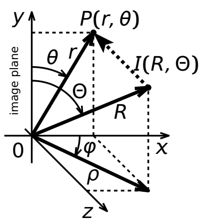
[Dribinski et al. (2002)]

Spherical/polar separable basis

$$I(R, \Theta) = \frac{I_{\text{tot}}(R)}{4\pi}[1 + \beta_1(R)P_1(\cos\Theta) + \beta_2(R)P_2(\cos\Theta) + \ldots] = \sum_n I_n(R)\cos^n\Theta, \quad I_n = \frac{I_{\text{tot}}}{4\pi}\sum_m a_{nm}\beta_m$$

Forward Abel transform:

$$\int I(R, \Theta)\,dz = \sum_n \int I_n(R)\cos^n\Theta\,dz = \sum_n \int I_n(R)\left(\frac{r}{R}\cos\theta\right)^n dz = \sum_n \underbrace{\left[\int I_n(R)\left(\frac{r}{R}\right)^n dz\right]}_{P_n(r)}\cos^n\theta =$$



$$R = \sqrt{r^2 + z^2} \qquad \cos\Theta = \frac{r}{R}\cos\theta$$

That is, the projected image is representable in the same form:

$$P(r, \theta) = \sum_n P_n(r)\cos^n\theta,$$

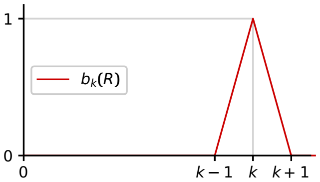
with the "projected" radial distributions

$$P_n(r) = \int I_n(R)\left(\frac{r}{R}\right)^n dz$$

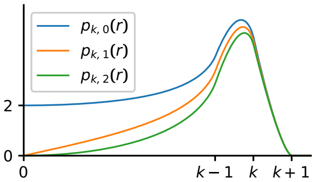
separably related to the initial radial distributions.

Radial basis functions

The radial distributions $I_n(R)$ can be approximated by continuous piecewise linear functions. Their space is spanned by a basis consisting of triangular functions:



$$b_k(R) = \begin{cases} R - (k-1), & R \in [k-1, k], \\ (k+1) - R, & R \in [k, k+1], \\ 0 & \text{otherwise.}\end{cases}$$



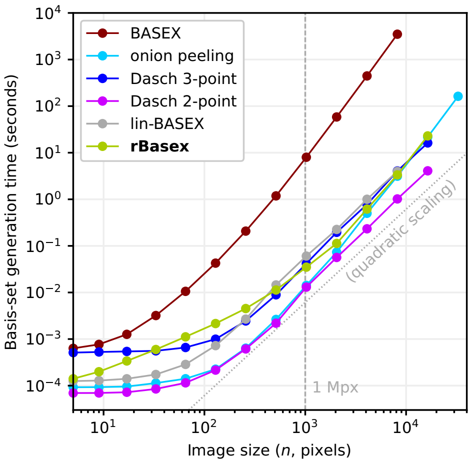
Their Abel transforms $p_{k,n}(r) = \int b_k(R)\left(\frac{r}{R}\right)^n dz$ have relatively simple *analytical* forms for any radial position k and angular order n.
[PyAbel, 2021]

Algorithm (inverse transform)

1. Extract the $\cos^n\theta$ radial profiles $P_n(r)$ from the input image.
2. Find their coordinates $c_{k,n}$ over the projected basis:
   $P_n(r) = \sum_k c_{k,n}p_{k,n}(r)$
   for each angular order n (separately).
3. The reconstructed $\cos^n\theta$ radial profiles are just
   $I_n(R) = \sum_k c_{k,n}b_k(R)$,
   and the reconstructed 3D distribution is
   $I(R, \Theta) = \sum_n I_n(R)cos^n\Theta$.
4. If needed, create the reconstructed image by evaluating this $I(R, \Theta)$ at each pixel.
5. If needed, convert $I(R, \Theta)$ to the
   $\frac{I_{\text{tot}}(R)}{4\pi}[1 + \beta_1(R)P_1(\cos\Theta) + \beta_2(R)P_2(\cos\Theta) + \ldots]$
   representation.

Forward transform: same idea, but with swapping the projected ($p_{k,n}$) and distribution ($b_{k,n}$) basis sets.

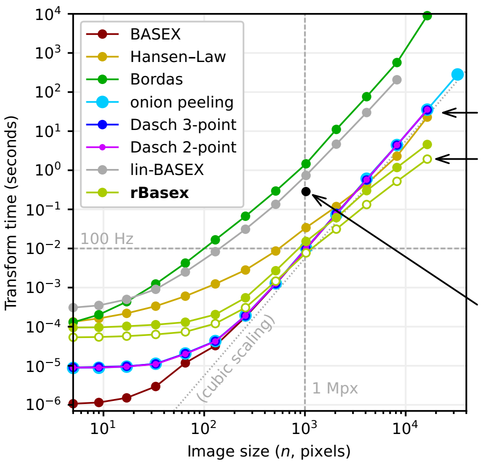
Basis-set precalculation time



(BASEX took minutes for 1 Mpx before PyAbel optimizations)

pBasex takes *hour(s)* for <1 Mpx

Transform time

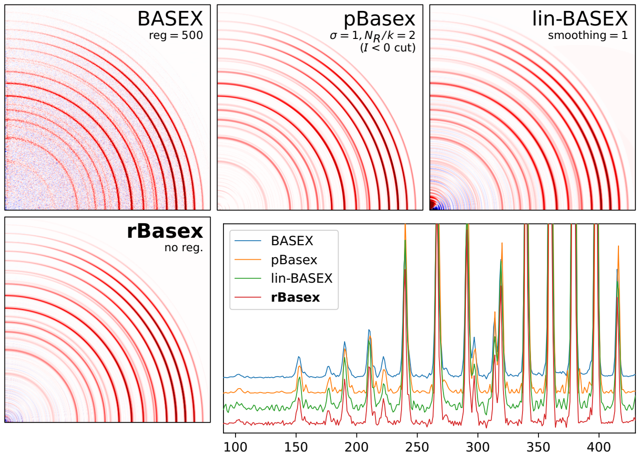


only image reconstruction (distributions add ~5 ms at 1 Mpx)

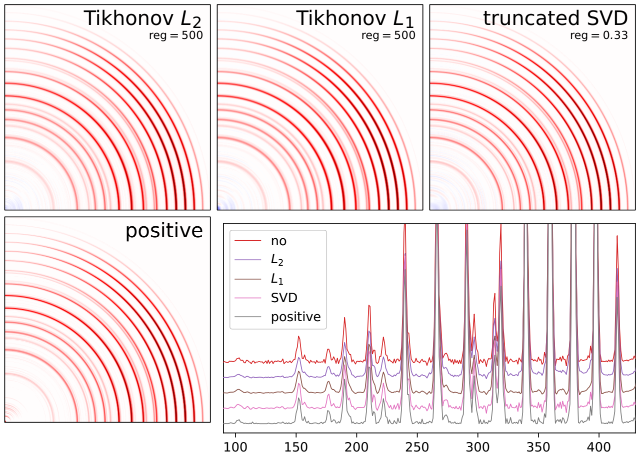
open circles: distributions only (without reconstructed image)

pBasex takes ~300 ms at 1 Mpx

Output quality



Regularizations



Summary of features

- Fast. Outperforms fastest available methods for images ≳ 1 megapixel.
- Low memory consumption.
- High resolution. Reliably resolves 1-pixel features.
- Any (even and odd) angular orders up to ∼ 15.
- Correct analysis of *partial* images by masking missing, obscured, damaged, contaminated parts.
- Can be easily extended to analyze *sliced* VMI data with *arbitrary* slicing-pulse thicknesses and shapes ([Ryazanov (2012)], not yet implemented in PyAbel).

Acknowledgment

Many thanks to my PyAbel colleagues
Daniel D. Hickstein and Stephen T. Gibson
for useful discussions, suggestions and testing!

References

VMI: D. W. Chandler, P. L. Houston, J. Chem. Phys. **87**(2), 1445 (1987). doi:10.1063/1.453276.

A. T. J. B. Eppink, D. H. Parker, Rev. Sci. Instrum. **68**(9), 3477 (1997). doi:10.1063/1.1148310.

BASEX: V. Dribinski, A. Ossadtchi, V. A. Mandelshtam, H. Reisler, Rev. Sci. Instrum. **73**(7), 2634 (2002). doi:10.1063/1.1482156.

Separable basis, sliced VMI: M. Ryazanov, *Development and implementation of methods for sliced velocity map imaging. Studies of overtone-induced dissociation and isomerization dynamics of hydroxymethyl radical (CH_2OH and CD_2OH)*, Ph.D. dissertation, University of Southern California, 2012.

pBasex: G. A. Garcia, L. Nahon, I. Powis, Rev. Sci. Instrum. **75**(11), 4989–4996 (2004). doi:10.1063/1.1807578.

lin-BASEX: Th. Gerber, Yu. Liu, G. Knopp, P. Hemberger, A. Bodi, P. Radi, Ya. Sych, Rev. Sci. Instrum., **84**(3), 033101 (2013). doi:10.1063/1.4793404.

L_1 regularization: K. J. Daun, K. A. Thomson, F. Liu, G. J. Smallwood, Appl. Opt. **45**(19), 4638 (2006). doi:10.1364/AO.45.004638.

PyAbel: D. D. Hickstein, S. T. Gibson, R. Yurchak, D. D. Das, M. Ryazanov, Rev. Sci. Instrum. **90**(6), 065115 (2019). doi:10.1063/1.5092635.

S. Gibson, D. D. Hickstein, R. Yurchak, M. Ryazanov, D. Das, G. Shih, *PyAbel/PyAbel: v0.8.4*. doi:10.5281/zenodo.4690660.

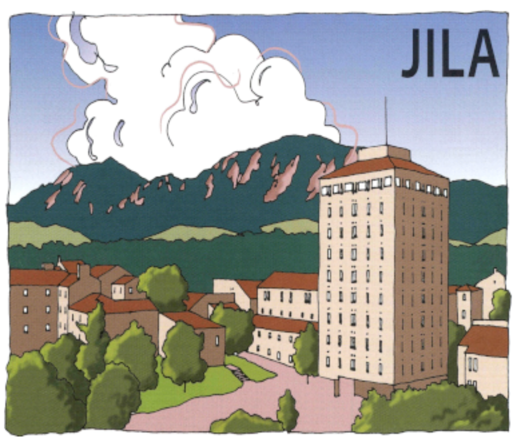
PyAbel at GitHub: `https://github.com/PyAbel/PyAbel`.

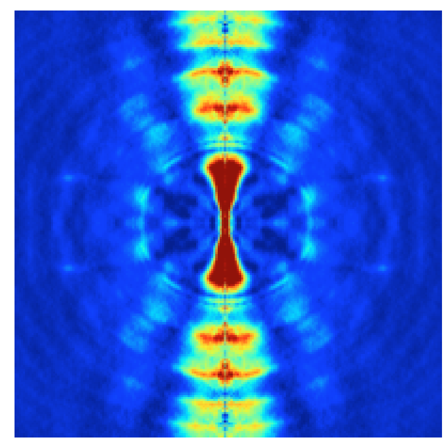
ISMB 2021

Fast high-resolution method for forward and inverse Abel transforms of velocity-map images

Mikhail Ryazanov

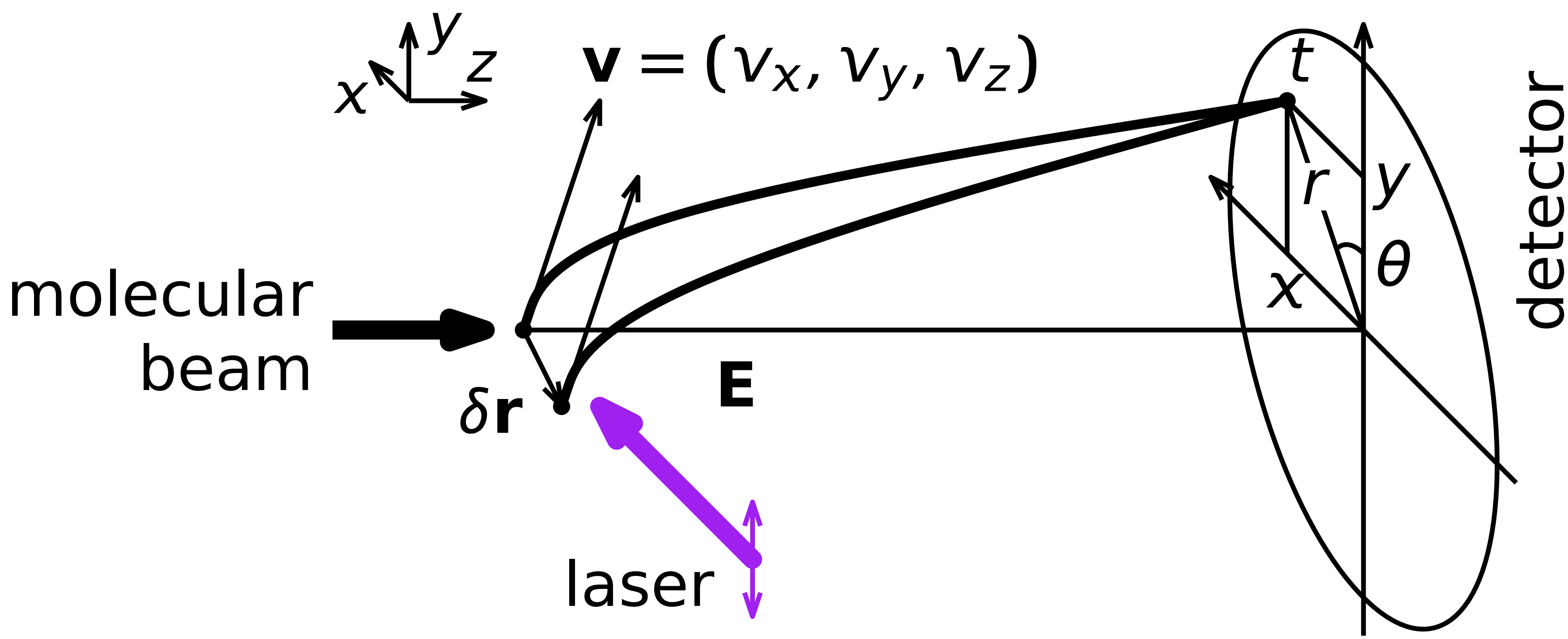
JILA, 440 UCB, Boulder, CO 80309, USA





PyAbel at GitHub

Velocity-map imaging



$\mathbf{v} \mapsto (x, y, t)$ [Chandler, Houston (1987)]
independent of $\delta\mathbf{r}$ [Eppink, Parker (1997)]

$$\begin{cases} (x, y) \approx \alpha_{\text{rad}} \cdot t_0 \cdot (v_x, v_y) \\ t \approx t_0 - \alpha_{\text{ax}} \cdot v_z \end{cases} \qquad \begin{cases} r \sim v_{\text{rad}} \\ t \sim v_{\text{ax}} + \text{const} \end{cases}$$

Cylindrical symmetry: $I(\mathbf{v}) = I(v, \theta)$
(axis ∥ polarization) $I(\theta) \sim P_n(\cos\theta)$

Abel transform

Cylindrically symmetric distribution

$$I(x, y, z) = I(\rho, y) \qquad \rho = \sqrt{x^2 + z^2}$$

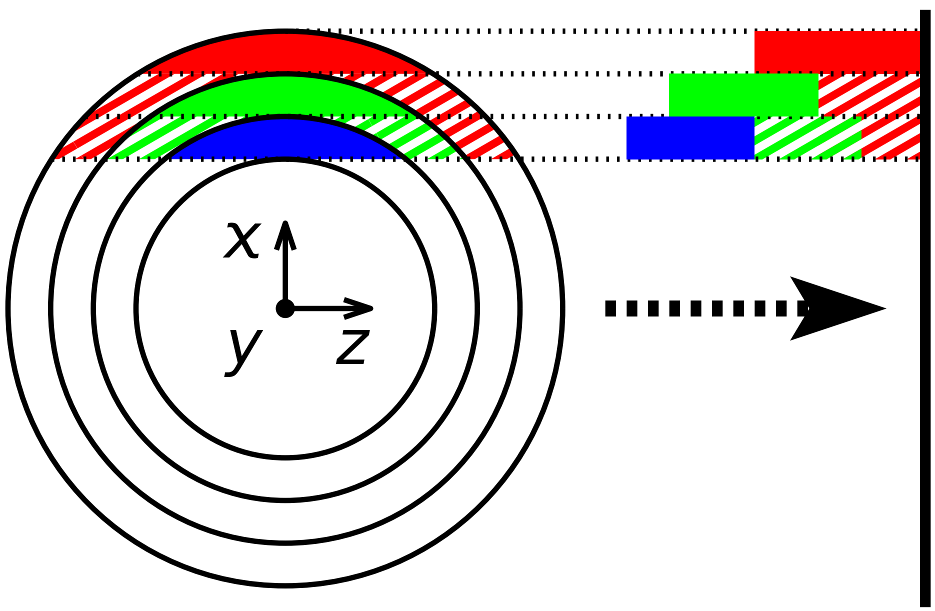
$$x = \rho \cos \varphi$$

$$z = \rho \sin \varphi$$

Forward transform is a parallel projection:

$$P(x, y) = \int_{-\infty}^{+\infty} I(x, y, z)\, dz = $$

$$= \int_{|x|}^{\infty} I(\rho, y) \frac{\rho\, d\rho}{\sqrt{\rho^2 - x^2}}$$



Inverse transform is formally

$$I(\rho, y) = -\frac{1}{\pi} \int_{\rho}^{\infty} \frac{dP(x,y)}{dx} \frac{1}{\sqrt{x^2 - \rho^2}}\, dx$$

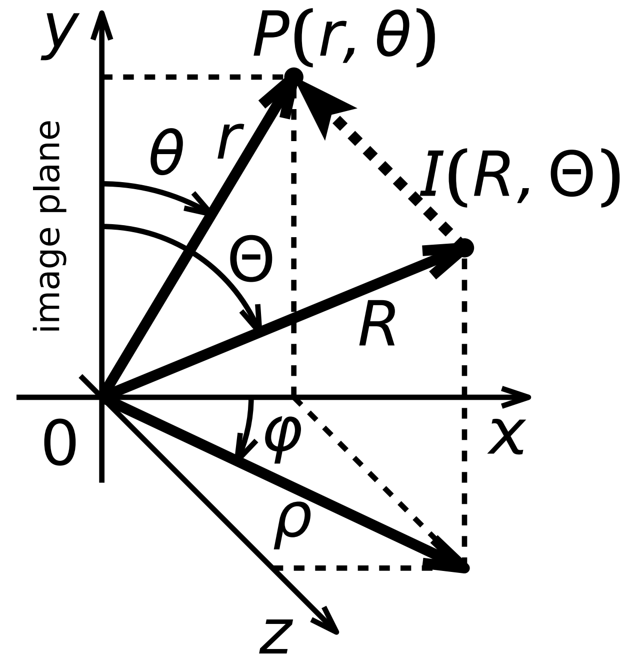
In practice: noisy singularity

- explicit basis set
- regularization

[Dribinski et al. (2002)]

Spherical/polar separable basis

$$I(R,\Theta) = \frac{I_\text{tot}(R)}{4\pi}\left[1 + \beta_1(R)P_1(\cos\Theta) + \right.$$
$$\left. + \beta_2(R)P_2(\cos\Theta) + \ldots\right] =$$
$$= \sum_n I_n(R)\cos^n\Theta, \qquad I_n = \frac{I_\text{tot}}{4\pi}\sum_m a_{nm}\beta_m$$



$$R = \sqrt{r^2 + z^2}$$
$$\cos\Theta = \frac{r}{R}\cos\theta$$

Forward Abel transform:

$$\int I(R,\Theta)\,dz = \sum_n \int I_n(R)\cos^n\Theta\,dz =$$
$$= \sum_n \int I_n(R)\left(\frac{r}{R}\cos\theta\right)^n dz =$$
$$= \sum_n \underbrace{\left[\int I_n(R)\left(\frac{r}{R}\right)^n dz\right]}_{P_n(r)}\cos^n\theta =$$

That is, the projected image is representable in the same form:

$$P(r,\theta) = \sum_n P_n(r)\cos^n\theta,$$

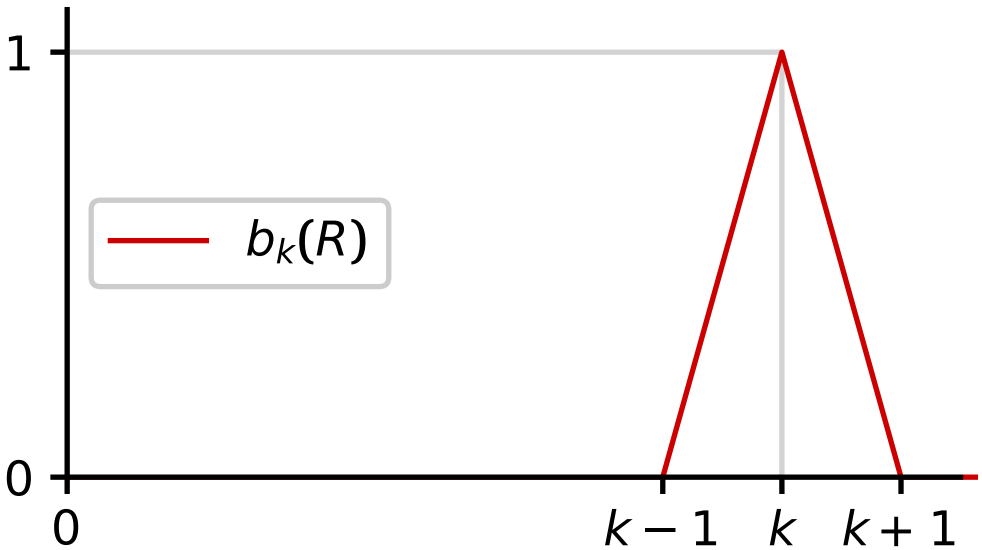
with the “projected” radial distributions

$$P_n(r) = \int I_n(R)\left(\frac{r}{R}\right)^n dz$$

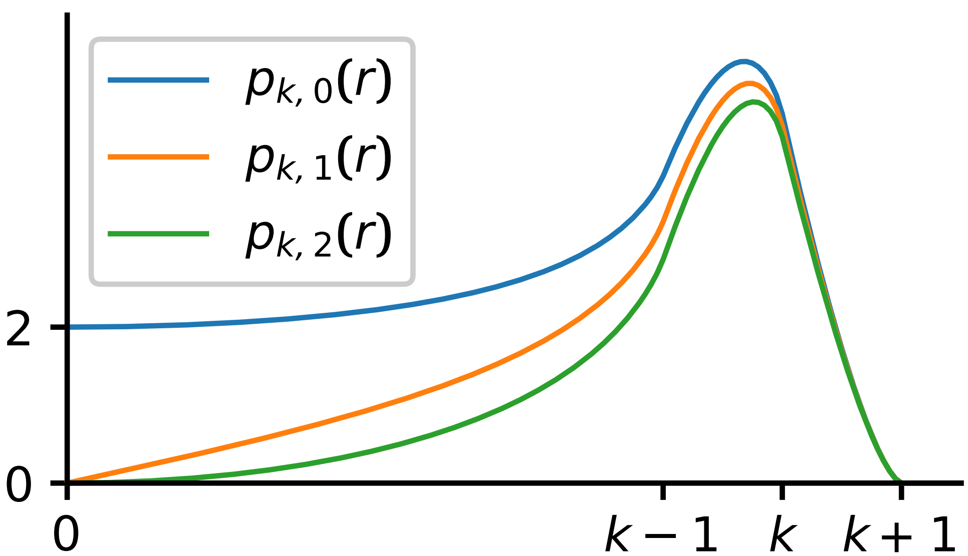
separably related to the initial radial distributions.

Radial basis functions

The radial distributions $I_n(R)$ can be approximated by continuous piecewise linear functions. Their space is spanned by a basis consisting of triangular functions:



$$b_k(R) = \begin{cases} R-(k-1), & R \in [k-1, k], \\ (k+1)-R, & R \in [k, k+1], \\ 0 & \text{otherwise.} \end{cases}$$



Their Abel transforms $p_{k,n}(r) = \int b_k(R) \left(\frac{r}{R}\right)^n dz$ have relatively simple *analytical* forms for any radial position k and angular order n.

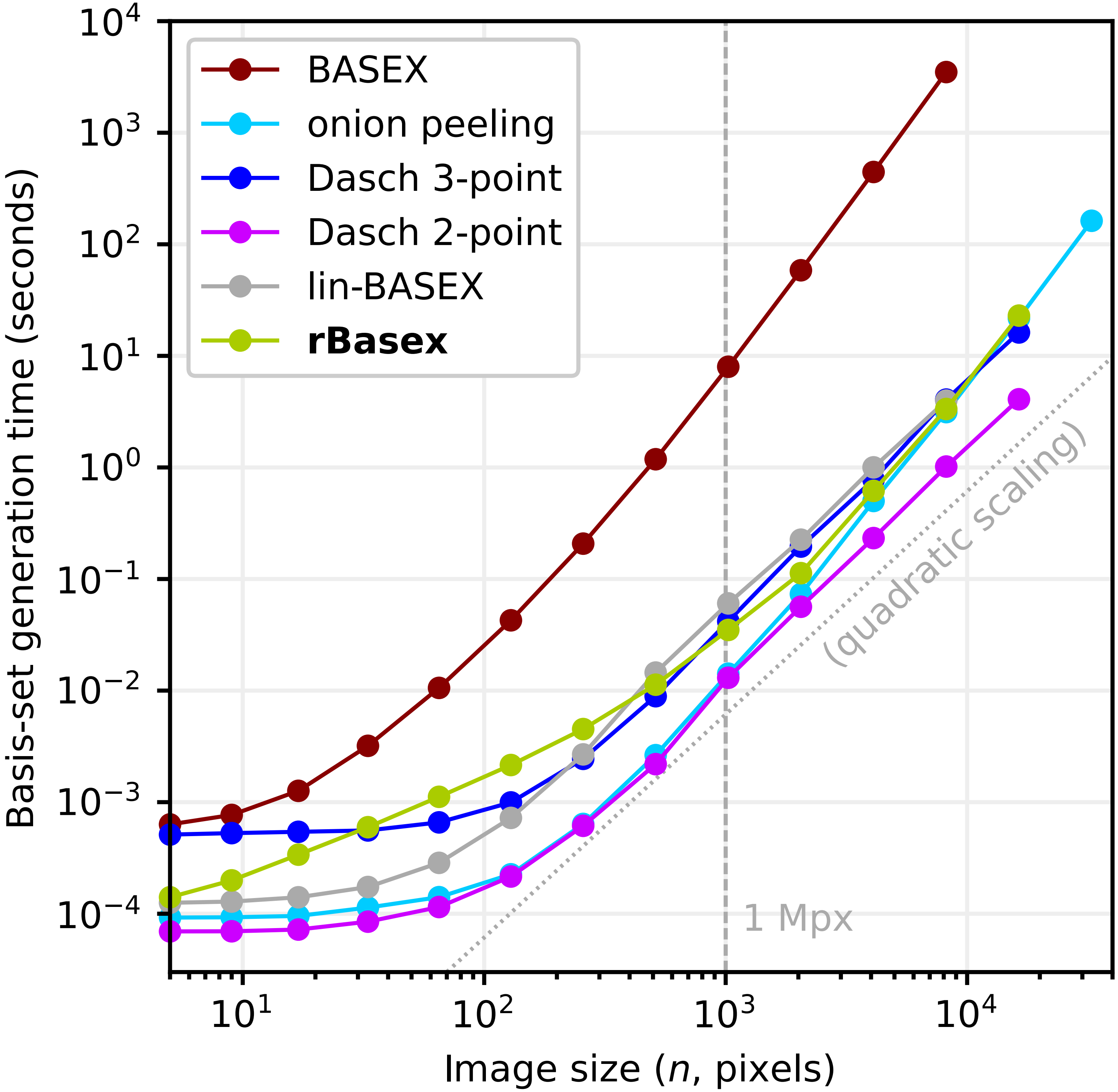
[PyAbel, 2021]

Algorithm (inverse transform)

1. Extract the $\cos^n\theta$ radial profiles $P_n(r)$ from the input image.
2. Find their coordinates $c_{k,n}$ over the projected basis:
   $$P_n(r) = \sum_k c_{k,n} p_{k,n}(r)$$
   for each angular order n (separately).
3. The reconstructed $\cos^n\theta$ radial profiles are just
   $$I_n(R) = \sum_k c_{k,n} b_k(R),$$
   and the reconstructed 3D distribution is
   $$I(R,\Theta) = \sum_n I_n(R) cos^n \Theta.$$
4. If needed, create the reconstructed image by evaluating this $I(R,\Theta)$ at each pixel.
5. If needed, convert $I(R,\Theta)$ to the
   $$\frac{I_{\text{tot}}(R)}{4\pi}[1 + \beta_1(R)P_1(\cos\Theta) + \beta_2(R)P_2(\cos\Theta) + \ldots]$$
   representation.

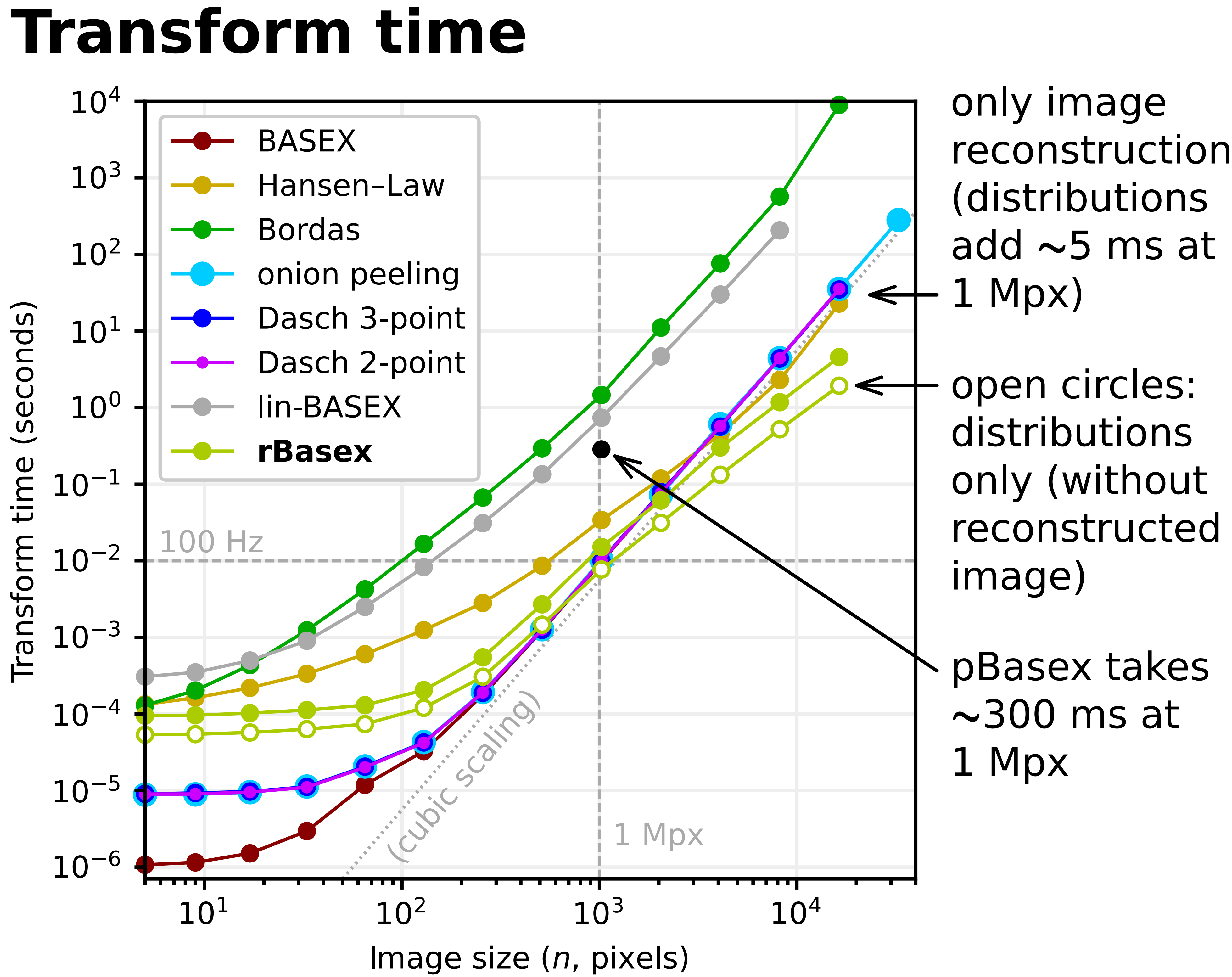
Forward transform: same idea, but with swapping the projected ($p_{k,n}$) and distribution ($b_{k,n}$) basis sets.

Basis-set precalculation time

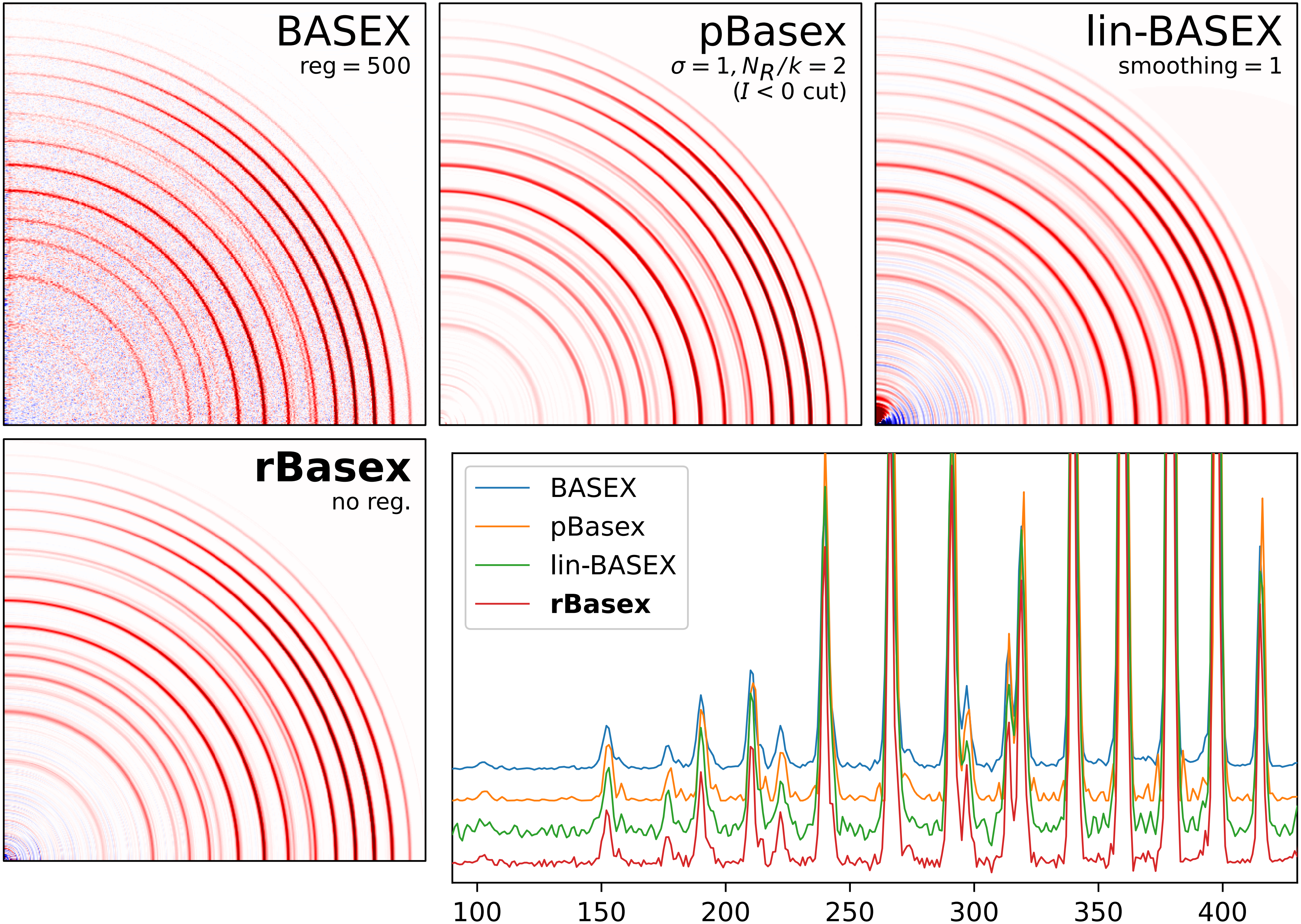


(BASEX took minutes for 1 Mpx before PyAbel optimizations)

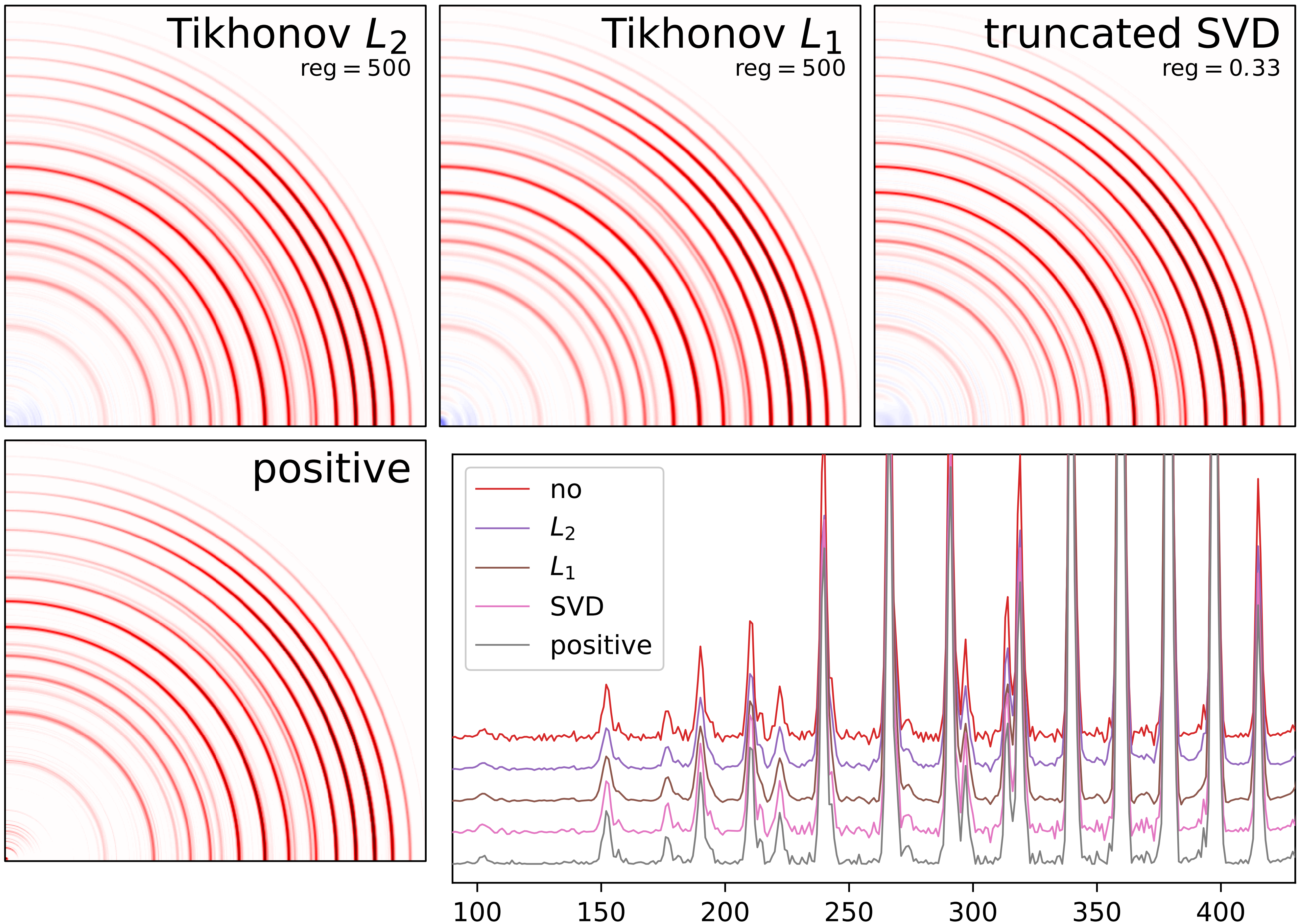
pBasex takes *hour(s)* for <1 Mpx


Transform time
Transform time (seconds)
Image size (n, pixels)
BASEX
Hansen–Law
Bordas
onion peeling
Dasch 3-point
Dasch 2-point
lin-BASEX
rBasex
100 Hz
1 Mpx
(cubic scaling)
only image reconstruction (distributions add ~5 ms at 1 Mpx)
open circles: distributions only (without reconstructed image)
pBasex takes ~300 ms at 1 Mpx

Output quality



Regularizations



Summary of features

- Fast. Outperforms fastest available methods for images $\gtrsim 1$ megapixel.
- Low memory consumption.
- High resolution. Reliably resolves 1-pixel features.
- Any (even and odd) angular orders up to ~ 15.
- Correct analysis of *partial* images by masking missing, obscured, damaged, contaminated parts.
- Can be easily extended to analyze *sliced* VMI data with *arbitrary* slicing-pulse thicknesses and shapes ([Ryazanov (2012)], not yet implemented in PyAbel).

Acknowledgment

Many thanks to my PyAbel colleagues
Daniel D. Hickstein and Stephen T. Gibson
for useful discussions, suggestions and testing!

References

VMI: D. W. Chandler, P. L. Houston, J. Chem. Phys. **87**(2), 1445 (1987). doi:10.1063/1.453276.

A. T. J. B. Eppink, D. H. Parker, Rev. Sci. Instrum. **68**(9), 3477 (1997). doi:10.1063/1.1148310.

BASEX: V. Dribinski, A. Ossadtchi, V. A. Mandelshtam, H. Reisler, Rev. Sci. Instrum. **73**(7), 2634 (2002). doi:10.1063/1.1482156.

Separable basis, sliced VMI: M. Ryazanov, *Development and implementation of methods for sliced velocity map imaging. Studies of overtone-induced dissociation and isomerization dynamics of hydroxymethyl radical (CH_2OH and CD_2OH)*, Ph.D. dissertation, University of Southern California, 2012.

pBasex: G. A. Garcia, L. Nahon, I. Powis, Rev. Sci. Instrum. **75**(11), 4989–4996 (2004). doi:10.1063/1.1807578.

lin-BASEX: Th. Gerber, Yu. Liu, G. Knopp, P. Hemberger, A. Bodi, P. Radi, Ya. Sych, Rev. Sci. Instrum., **84**(3), 033101 (2013). doi:10.1063/1.4793404.

L_1 regularization: K. J. Daun, K. A. Thomson, F. Liu, G. J. Smallwood, Appl. Opt. **45**(19), 4638 (2006). doi:10.1364/AO.45.004638.

PyAbel: D. D. Hickstein, S. T. Gibson, R. Yurchak, D. D. Das, M. Ryazanov, Rev. Sci. Instrum. **90**(6), 065115 (2019). doi:10.1063/1.5092635.

S. Gibson, D. D. Hickstein, R. Yurchak, M. Ryazanov, D. Das, G. Shih, *PyAbel/PyAbel: v0.8.4*. doi:10.5281/zenodo.4690660.

PyAbel at GitHub: `https://github.com/PyAbel/PyAbel`.