CCbeyond =
Continuous Charge Approach, Self Consistent, 2D

continuous charge distributions in non-uniform motion that change shape
EM fields and motion

application: bunch compressors for FELs

why not wake codes (conventional time domain solvers)?

problem domain is too large even with window techniques

sources in free space, chamber boundaries cannot be considered
(echoes from boundaries do not catchup or influence weakly)

Trafic4

together with Andreas Kabel

uses 3d sub bunches, each with its own 3d trajectory

sub bunch shape is fixed, motion is translational

trajectories are calculated by tracking of test particles in the center of sub bunches
motion is influenced by external and self fields

each trajectory has to be stored

Fortran and C; needs **parallel computing**; not available any more

it was one of the first **proof-of-principle program**s
number of sub-bunches rather limited
simulation of setups quite **time consuming**

↓

development of 1d-CSR method (see DESY-TEMF seminar December 2024)
empirical, utilizes radiation effects
numerically efficient, scalar computers
production runs (e.g. optimization of parameters)

CSRtrack

realizes both methods:

expensive parallel proof-of-principle sub-bunch method or
efficient scalar 1d method

setups with **2d reference** trajectory

trajectories are **not stored**; they are calculated by **back-tracking** from the actual particle position & momentum **in external fields**; that's a pretty good approximation for the recent past;

restarts are no problem

Fortran; sub-bunch method needs **parallel computing**; source codes exists, is frozen!

CSRtrack is a stand alone program that was used in combination with **Astra** and sometimes **Elegant** to perform “start-to-end” simulations; Igor Zagorodnov's **GlueTrack** automated the interaction, file formats, and launch of individual codes; this was the state of art for early simulations of TTF (later Flash) and the European XFEL
it was the first program with 1d-CSR in more than just one magnet + drifts

↓

development of all-in-one programs as **Xtrack** and **Ocelot**

Xtrack

s2e program with many effects (e.g. SC, CSR, wakes, cavities+CK, Laser heater, …)
but not the gun ($\rightarrow$ Astra&Co) and not the FEL ($\rightarrow$ Genesis or Alice)
scalar 1d-CSR method for fully 3d reference trajectories

remark: 1d-CSR is equivalent to transient monopole wake; however, the wake is anti-causal and is determined by the preceding reference trajectory

Matlab; frozen! all methods are available in Ocelot, which is written in Python
please use Ocelot

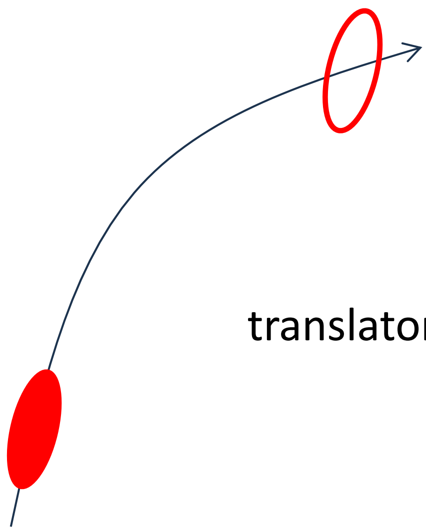
$\downarrow$

the **need to verify** empirically efficient methods **still exists**, especially as increasingly aggressive operating points are being approached

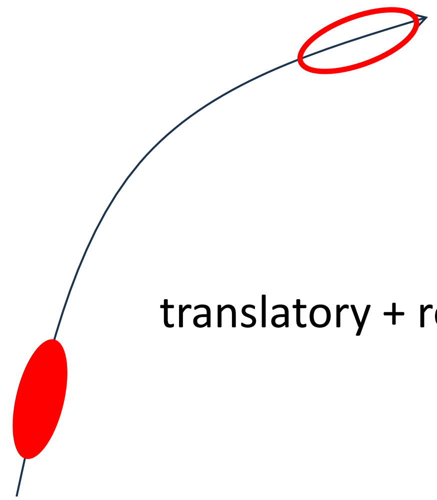
since CSRtrack's sub-bunch method has its limitations (number of sub-bunches and resolution) and the program is almost no longer available and is not being adapted to new computers, the desire for an **alternative** arose

development of CC and CCbeyond

Sub-Bunch



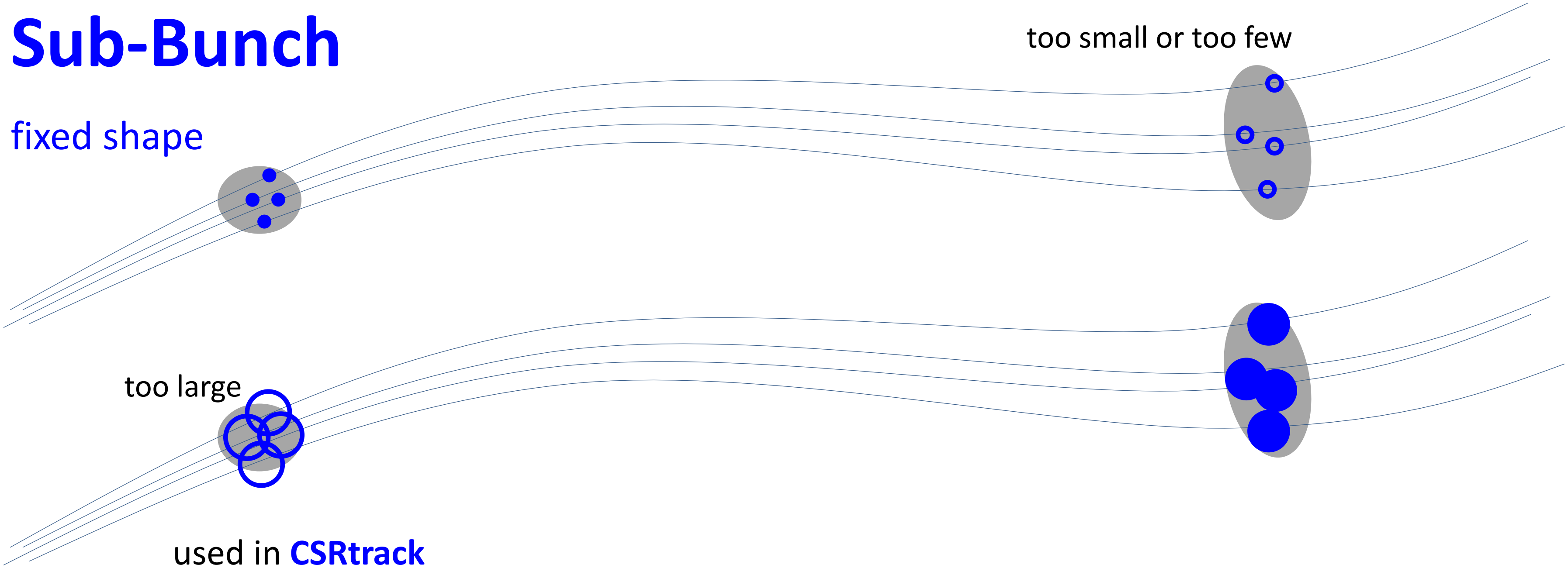
translatory



translatory + rotation ?

Sub-Bunch

fixed shape



used in **CSRtrack**

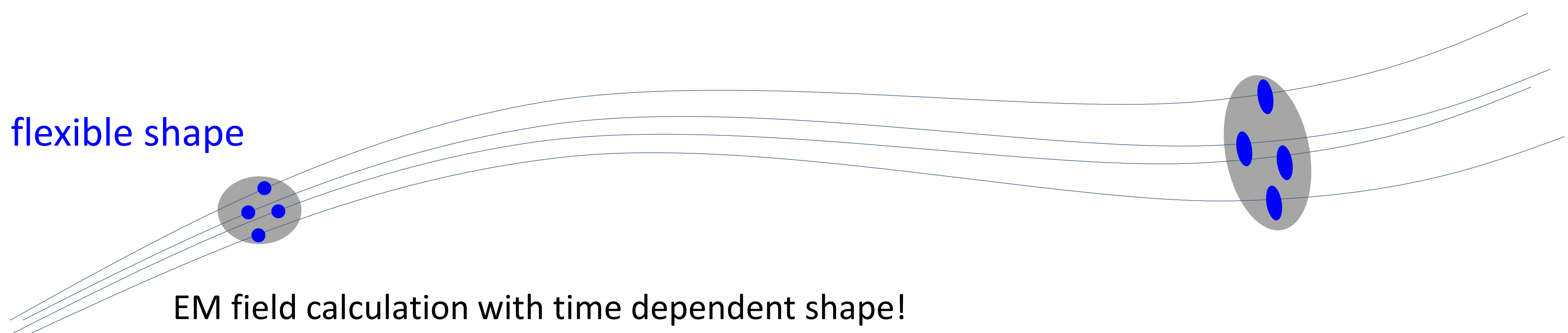
size of sub-bunches is determined by smallest beam dimension

number of sub-bunches is determined by maximal ratio of smallest/largest dimension

the fixed shape method is **equivalent to low pass filtering** of sources or fields

adaptive fixed shape is possible but limited by interaction length

flexible shape



EM field calculation with time dependent shape!

CC has a single flexible bunch with dimensions defined by optics

Continuous Charge

point particles (N macro particles) $\xrightarrow{N \to \infty}$ continuous source

$\mathbf{r}_\nu$, $\mathbf{p}_\nu$, $q_\nu = Q/N$, $\nu = 1 \ldots N$

$d/dt\, [\mathbf{r}_\nu, \mathbf{p}_\nu] = [\mathbf{v}_\nu, \mathbf{F}_\nu]$ **Newtons's EoM**

$f(\mathbf{r}, \mathbf{p}, t)$ phase space density

$d/dt\, f(\mathbf{r}, \mathbf{p}, t) = 0$ **Vlasov Equation**

$\downarrow$ sources in geom. space

$$\rho(\boldsymbol{r}, t) = Q \int f(\boldsymbol{r}, \boldsymbol{p}, t) d\boldsymbol{p}$$

$$\boldsymbol{J}(\boldsymbol{r}, t) = Q \int \boldsymbol{v} f(\boldsymbol{r}, \boldsymbol{p}, t) d\boldsymbol{p}$$

CC

existing, GUI in preparation

Gaussian phase space in accelerator coordinates S, x

flat source (no extension in Z direction)

Z symmetric external magnetic field with $\hat{\boldsymbol{B}}(S,x,0) = B_Z\big(X(S,x), Y(S,x)\big)\boldsymbol{e}_Z$
only **linear optics**

unperturbed motion of source, effects to beam by perturbation

flat Gaussian source, with S dependent rms parameters

a detailed **comparison** of different methods with CC was presented in this seminar **last year**

CCbeyond

work in progress

initial arbitrary phase space in cartesian coordinates X, Y

flat source

Z symmetric external magnetic field with $\boldsymbol{B}(X,Y,0) = B_Z(X,Y)\boldsymbol{e}_Z$
with **non linear optical effects**

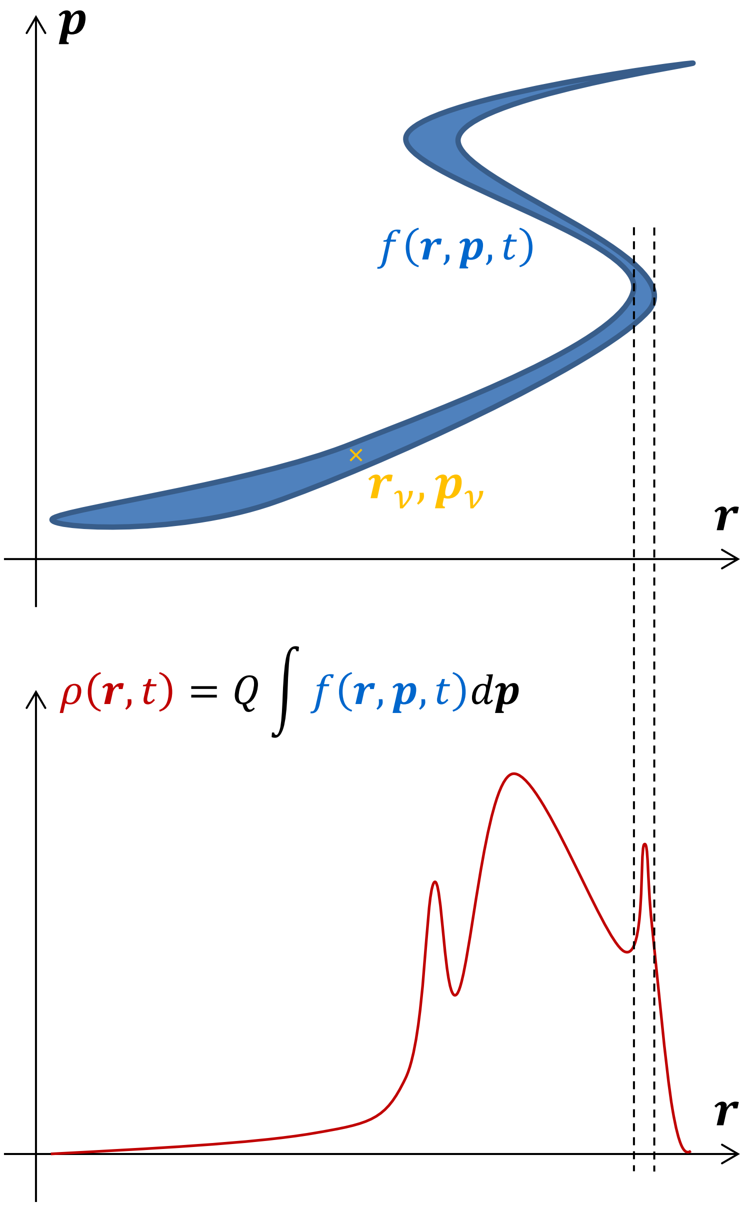
self consistent motion

flat arbitrary source

$$Q(\boldsymbol{r},t) = \tilde{Q}\left(\begin{pmatrix} X \\ Y \end{pmatrix}, t\right)\delta(Z)$$

Vlasov Solver

$$\frac{d}{dt}\mathrm{f}(\mathbf{r},\mathbf{p},\mathrm{t}) = 0 \qquad \frac{d}{dt}[\boldsymbol{r}_\nu, \boldsymbol{p}_\nu] = [\boldsymbol{v}_\nu, \boldsymbol{F}_\nu]$$



density is 4(6) dim object
needs to be projected to 2(3) dim
phase space points can be tracked as particles
phase space volume is conserved

a first approach: “**Monte Carlo Vlasov Solver**”
see G. Bassi, TUZBC03, PAC 2007

smoothing:

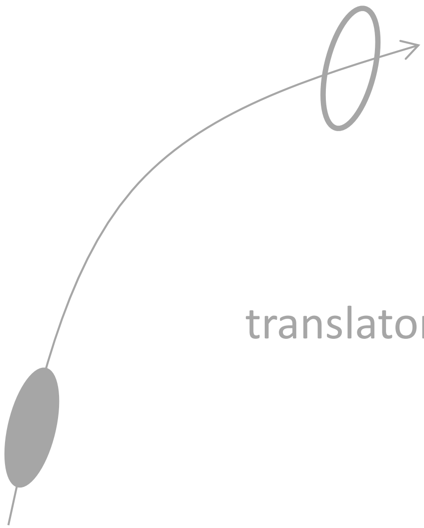
$$\mathrm{f}(\mathbf{r},\mathbf{p},\mathrm{t}) = \frac{1}{N}\sum b\big(\boldsymbol{r}-\boldsymbol{r}_v(t)\big)\delta\big(\boldsymbol{p}-\boldsymbol{p}_v(t)\big)$$

$$\rho(\mathbf{r},\mathrm{t}) = \frac{Q}{N}\sum b\big(\boldsymbol{r}-\boldsymbol{r}_v(t)\big)$$

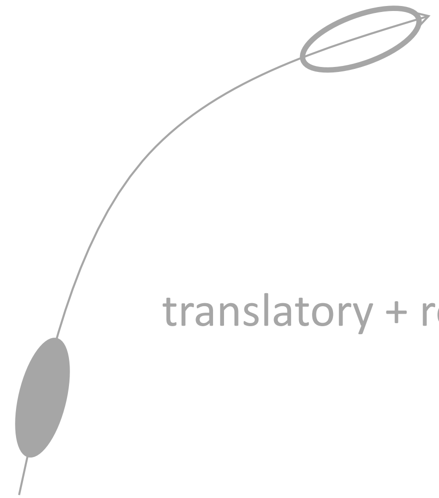
$$\boldsymbol{J}(\mathbf{r},\mathrm{t}) = \frac{Q}{N}\sum \boldsymbol{v}(\boldsymbol{p}_v)b\big(\boldsymbol{r}-\boldsymbol{r}_v(t)\big)$$

with a 2(3)d base function $b(\boldsymbol{r})$
the difference to sub-bunch approach
is of numerical nature

Sub-Bunch



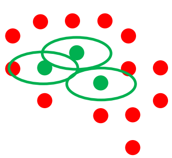
translatory

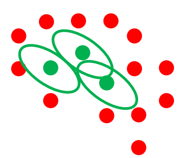


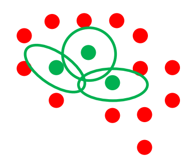
translatory + rotation ?

smoothing

$b(\boldsymbol{r})$







point distance << $\sigma_{smoothing}$ << σ_{macro}

?

N_p has to be large enough

Field Solver

fields in free space

$$\left(\nabla^2 - \frac{1}{c^2}\frac{\partial^2}{\partial t^2}\right) V = -Q \qquad V(\boldsymbol{r}_o, t) = \frac{1}{4\pi}\int \frac{Q(\boldsymbol{r}', t')}{\|\boldsymbol{r}_o - \boldsymbol{r}'\|} dV' \qquad t' = t - \|\cdots\|/c$$

sources $Q = \begin{bmatrix} \begin{bmatrix} \rho/\varepsilon \\ \mu J_X \\ \mu J_Y \end{bmatrix} \\ \partial_X [\,] \\ \partial_Y [\,] \\ \partial_t [\,] \end{bmatrix}$ potentials & derivatives $V = \begin{bmatrix} \begin{bmatrix} V_s \\ A_X \\ A_Y \end{bmatrix} \\ \partial_X [\,] \\ \partial_Y [\,] \\ \partial_t [\,] \end{bmatrix} \longrightarrow$ fields $\begin{aligned} E_X &= -\partial_X V - \partial_t A_X \\ E_Y &= \cdots \\ B_Z &= \cdots \end{aligned}$

(12 components)

flat (2d) source $Q(\boldsymbol{r}, t) = \tilde{Q}\left(\begin{pmatrix} X \\ Y \end{pmatrix}, t\right)\delta(Z)$

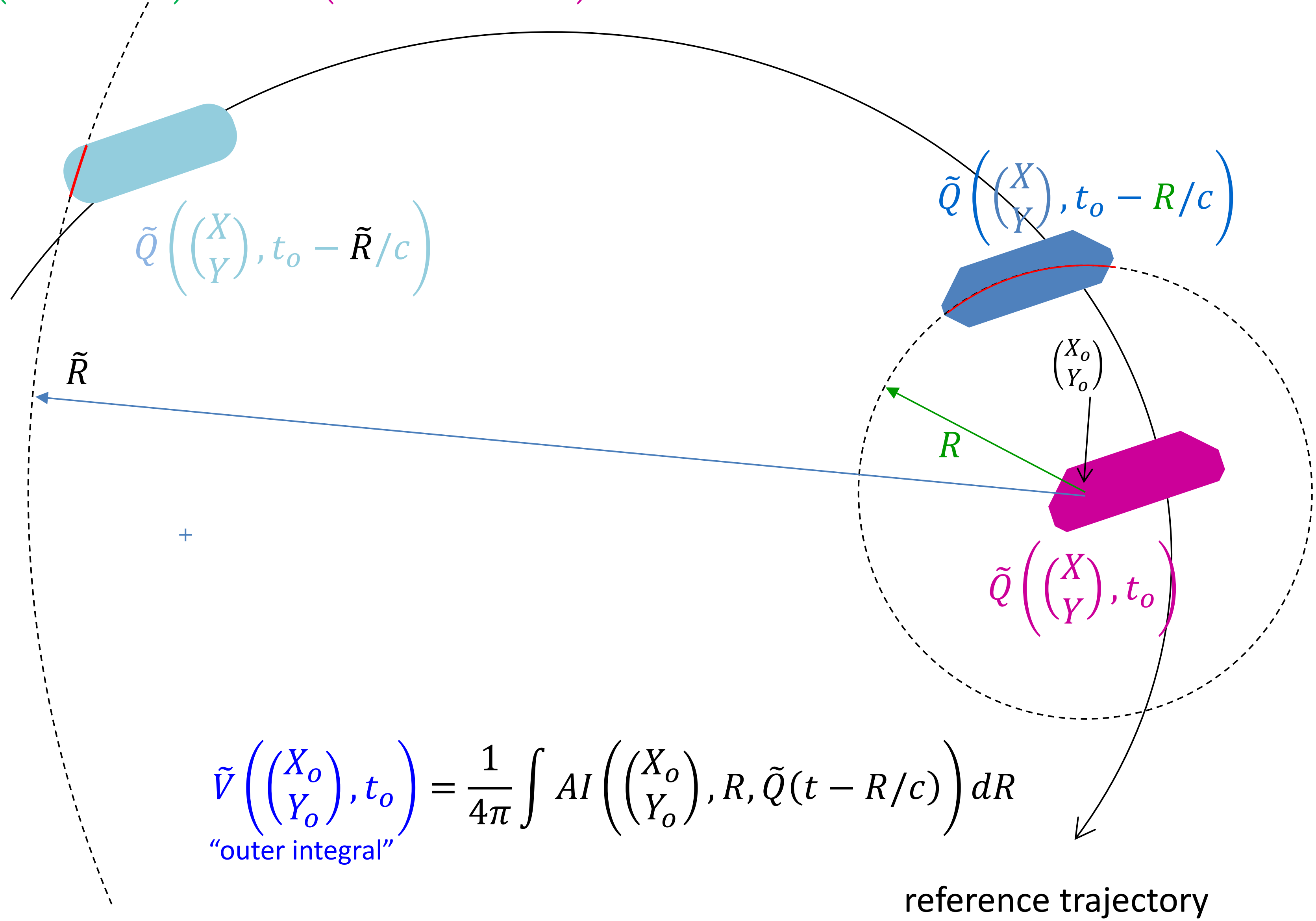
$$AI\left(\begin{pmatrix} X_o \\ Y_o \end{pmatrix}, R, \tilde{Q}\right) = \int \tilde{Q}\left(\begin{pmatrix} X_o - R\cos\varphi \\ Y_o - R\sin\varphi \end{pmatrix}\right) d\varphi \qquad \text{arc integral}$$

$$\tilde{V}\left(\begin{pmatrix} X_o \\ Y_o \end{pmatrix}, t\right) = \frac{1}{4\pi}\int AI\left(\begin{pmatrix} X_o \\ Y_o \end{pmatrix}, R, \tilde{Q}(t - R/c)\right) dR \qquad \text{outer integral}$$

see again G. Bassi, TUZBC03, PAC 2007

“arc integral”

$$AI\left(\begin{pmatrix} X_o \\ Y_o \end{pmatrix}, R, \tilde{Q}\right) = \int \tilde{Q}\left(\begin{pmatrix} X_o - R\cos\varphi \\ Y_o - R\sin\varphi \end{pmatrix}\right) d\varphi$$

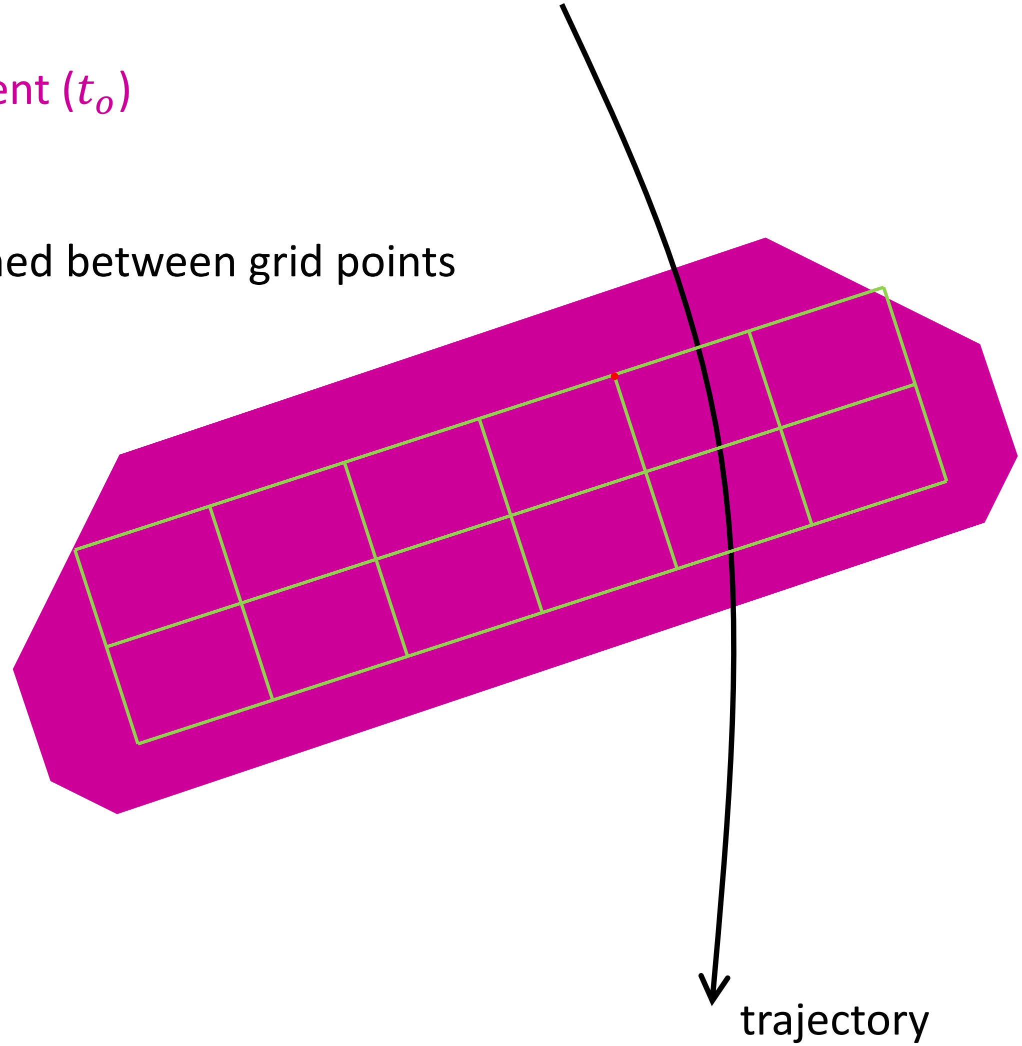


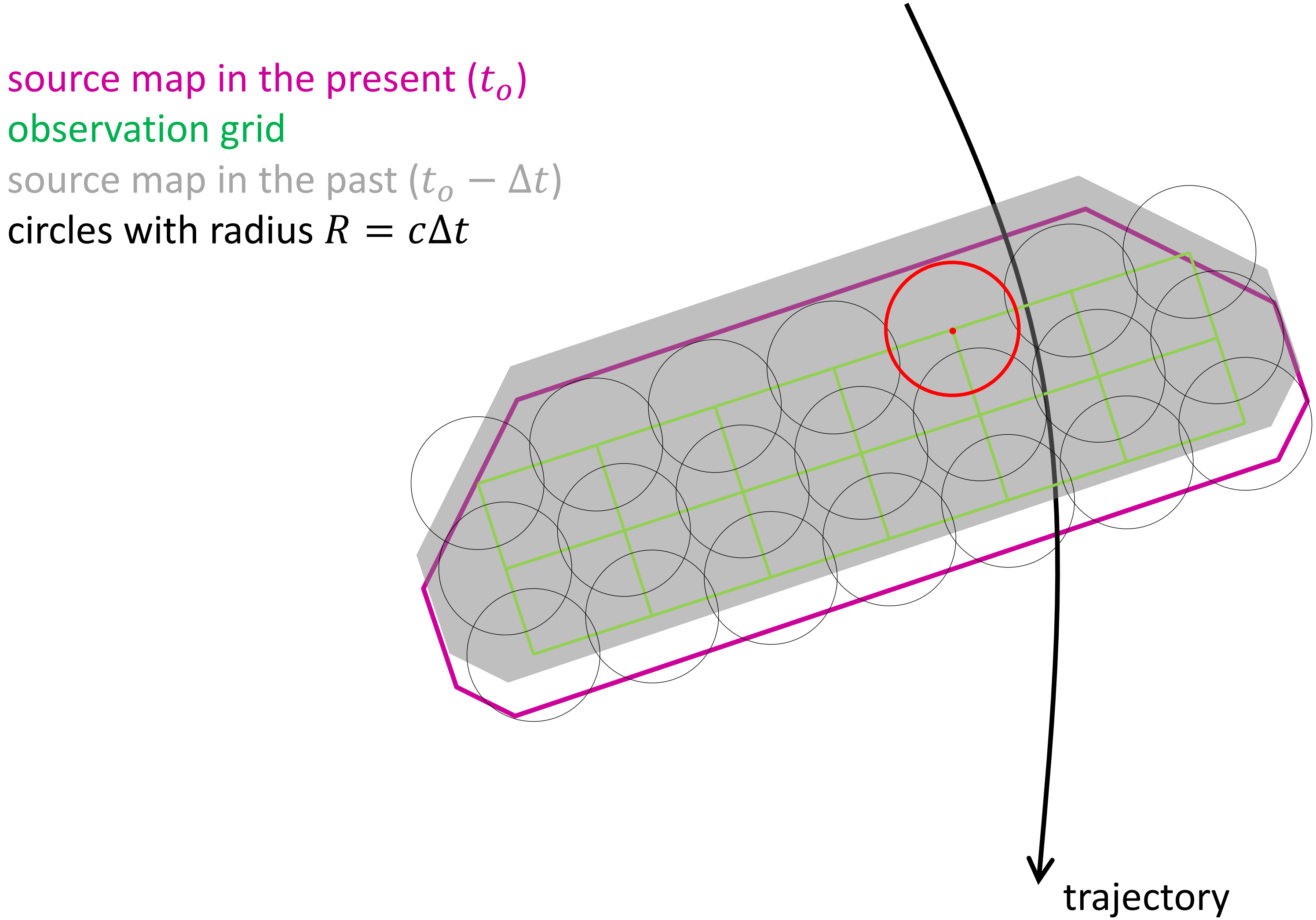
$$\tilde{V}\left(\begin{pmatrix} X_o \\ Y_o \end{pmatrix}, t_o\right) = \frac{1}{4\pi} \int AI\left(\begin{pmatrix} X_o \\ Y_o \end{pmatrix}, R, \tilde{Q}(t - R/c)\right) dR$$

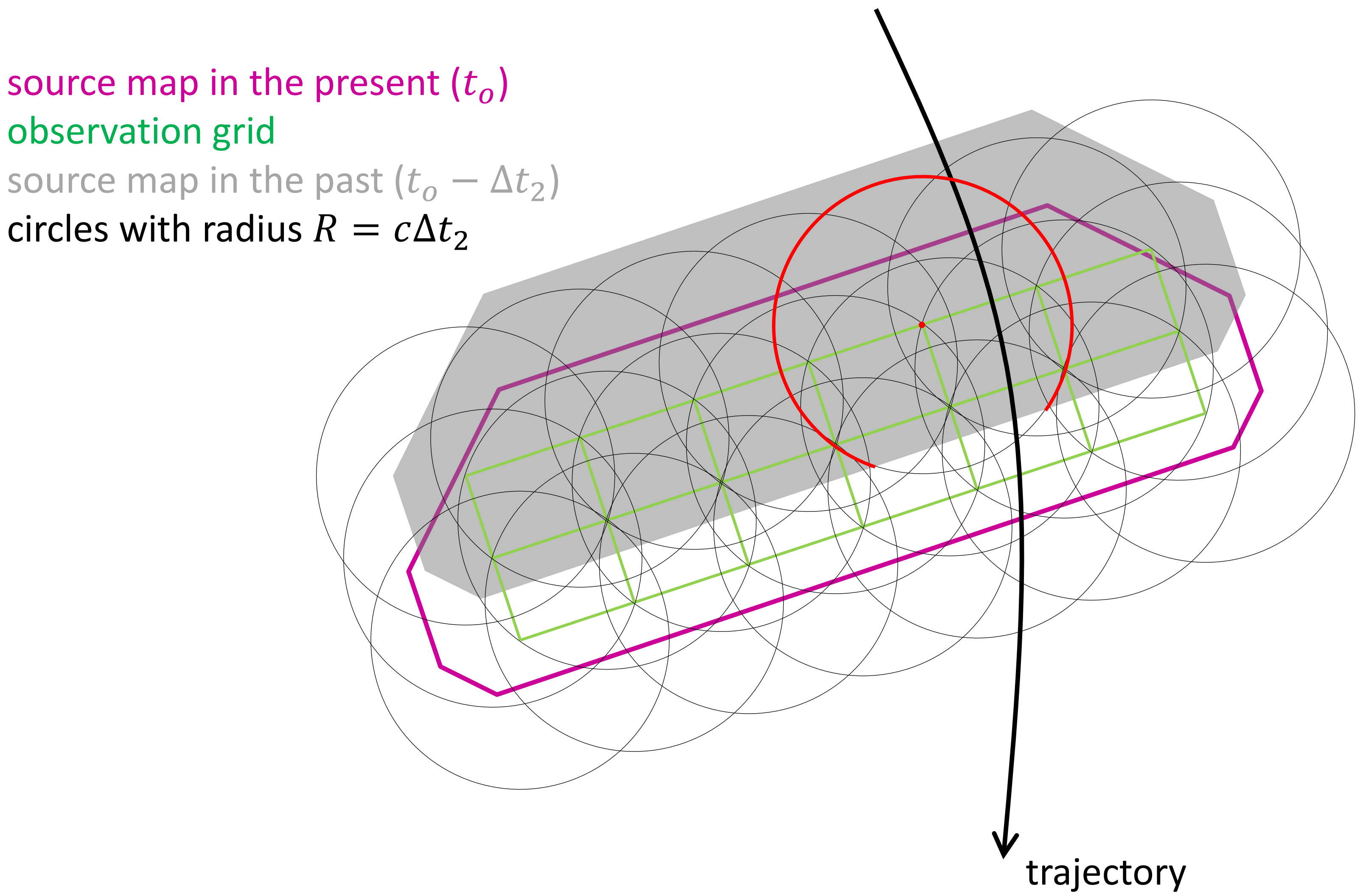
“outer integral”

source map in the present (t_o)
observation grid

interpolation is performed between grid points




source map in the present (t_o)
observation grid
source map in the past ($t_o - \Delta t$)
circles with radius $R = c\Delta t$
trajectory


source map in the present (t_o)
observation grid
source map in the past ($t_o - \Delta t_2$)
circles with radius $R = c\Delta t_2$
trajectory

Some Problems, Work in Progress

version 1
version 2

preliminary Vlasov solver

mapping

- Q-discretization (very flexible, but how to use degrees of freedom)
- observation grid
- arc- and projection integrals

stepping

- for equation of motion
- for recording of maps (history buffer)
- for outer integration (needs pre- and near-history)

validity and range of approximations

- pre-history (collective or individual uniform motion?)
- history (= mapping on track steps)
- near-history
- arc-or-projection decision

tracking and external fields

vertical offset

computational requirements

- matlab with parfor loop
- one or few days (BC2, about 10 CPUs)

first run soon