Is there a future for the OPA code?

BESSY, Berlin, April 3, 2025

Andreas Streun

- 1992-2022 PSI Villigen, Switzerland
- Lattice design work for SLS, CoSAMI, SLS 2.0 etc.
- Development of OPA lattice design code since 1993

about OPA

OPA Icon:
visualization of
storage ring
beam dynamics

- Design of [electron] storage rings (synchrotrons) and transfer lines
  - highly specialized task: global but small user community
- **Pros**
  - interactive and visual for quick and intuitive design from scratch
  - reasonably user friendly (?)
  - pragmatic: accelerator physics driven, developed during real design
  - fast: algorithms and implementations optimized for efficiency
- **Cons**
  - “one man show” (AS)
  - old fashioned programming style (1980’s origin):
    - not well structured (physics - graphics - data handling – I/O)
    - partially “spaghetti code”
    - not “object oriented”

Features

Accelerator Physics

- linear optics design with matching
- betatron coupling included
- non-linear dynamics optimization (NLO) with sextupoles and octupoles
- editors for longitudinal gradient bends and RF bucket
- several tracking modes
  - phase space and amplitude dependent tunes
  - DA grid probing, binary search and flood fill
  - Fast Touschek tracking (FTT)
- Orbit correction with girders and correlated misalignments
- Injection simulation and kicker timing
- Accelerator layout incl. geometric matching

Non-physics

- building block and text based lattice editors
- reads *and writes* lattice text files
  - partial import/export of ELEGANT, TRACY-2/-3, MAD, BMAD lattice files
- nice and well comprehensible plots
- EPS export
- works in Windows (W10) and Linux (xubuntu)
- fairly well documented

Loose ends

- Linear matching and NLO with coupling
- Tracking with errors
- 3D lattice geometry
- Catch and report all user errors
- Bug fixes… (Linux event handling problems)

History and contributors

- ~ mid 1980s: OPTIK code by Klaus Wille (DELTA)
  Borland Turbo Pascal on MS-DOS
- 1993 given to AS, modified and extended during SLS-design and named OPA
  Non-linear optimization and signal processing algorithms provided by Johan Bengtsson
  J. Bengtsson, The sextupole scheme for the Swiss Light Source (SLS): an analytic approach, SLS-Note 9/97 (1997)
- 200x ported to Borland Delphi on Windows
  Non-linear terms checked and shared with ELEGANT by Chun-xi Wang and Michael Borland.
  Octupoles and longitudinal gradient bends added in collaboration with Simon Leemann.
  S.C.Leemann and AS, Perspectives for future light source lattices incorporating yet uncommon magnets, Phys. Rev. ST Accel. Beams 14, 030701 (2011)
- 2019 ported to Lazarus Free Pascal (open source) on Windows and Linux
  Linear coupling and coupled radiation integrals added in collaboration with Volker Ziemann
  V. Ziemann and AS, Equilibrium parameters in coupled storage ring lattices and practical applications, Phys. Rev. Accel. Beams 25, 050703 (2022)
- July 2022 retirement AS, private OPA page https://andreas-streun.de/opa/
  Flood fill tracking added in collaboration with Bernard Riemann
  M. Aiba, J. Kallestrup, B. Riemann and AS, Efficient algorithms for dynamic aperture and momentum acceptance calculation in synchrotron light sources, Phys. Rev. Accel. Beams 27, 094002 (2024)

OPA Lattice Design Code

The main purpose of the OPA code is to support the development of electron (positron) storage rings. Emphasis is on visualization and interactivity rather than on elaborate beam dynamics models. OPA is in particular useful for designing high brightness light source lattices, but may be used for transfer lines and other types of lattices as well. Storage ring design with OPA starts from scratch and ends at a bare (i.e. error free) lattice with optimized dynamic apertures, to be passed on to other codes like TRACY, MAD or ELEGANT, which use more complete models.

OPA 4

Documentation

- **OPA user guide** (pdf)
- **"inside OPA"** (pdf): the physics part.
- **Tutorial** (pdf) (and example file from tutorial)

Homepage

https://andreas-streun.de/opa/

Downloads

- **Windows executable 4.062 zip**
- **Source files 4.062 zip** for Lazarus 2.2.0, work both for W10 and Ubuntu.
- A .deb file to install OPA on linux is not yet available (I don't know how to create one, help appreciated), instead you may download the source and install Lazarus to build the executable.
- Document sources user guide (tex.zip), "inside" (tex.zip), tutorial (docx).

Major changes from OPA version 3 to 4

- Transfer from Borland Delphi to Lazarus Free Pascal.
- Include coupling by Edward-Teng formalism.
- Also available for Linux now: tested for xubuntu 20.04 (some debugging going on...)

OPA 3.91d (2017)

The last (more or less) robust Delphi-version, for flat lattices only (no coupling).

Take a **Tour of OPA** to see if it may be useful for you too!

Read the **User guide (PDF)** to learn about its capabilities and limitations, how to install and use it and also about some physics behind.

Download the ZIP-file containing the program including example files and user guide. The program is a single executable and should run on any Microsoft Windows system.

This page is a mirror of the offical OPA page of the SynBD (former ADOS) group at PSI. It may be ahead in updates since the official page is not maintained continuously anymore due to my retirement.

Last update Mar 7, 2025. Andreas Streun

Demonstrations

- how to start from scratch
  - elements, cells, chromaticity
  - dynamic aperture, layout
- from step-by-step tutorial
  (available at the OPA homepage.) ➔
  - variables, coupling
  - injection, tracking
- elaborate design: SLS 2.0
  - orbit correction
  - nonlinear optimization
- eps export

OPA Tutorial (4.030)

Andreas Streun, PSI, January 2022

Let's design a small storage ring in order to get started with OPA and demonstrate its features!

This is OPA's start panel:

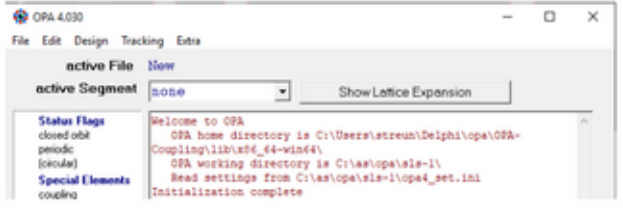


Fig.1 OPA start panel

File is for reading and saving .opa files and for exporting files to other codes.

Edit contains two editors, a plain text editor, and the interactive OPA Editor.

Design is the main part of OPA leading to modules for linear lattice design in terms of beta functions, non-linear optimization, orbit correction etc.

Tracking is for simulation of dynamic aperture, beam lifetime etc. in order to test the performance of the lattice.

Extra contains special options, some of them implemented preliminary or temporary only.

Editor

We don't load an existing .opa file but start from scratch, by using Edit → OPA Editor:

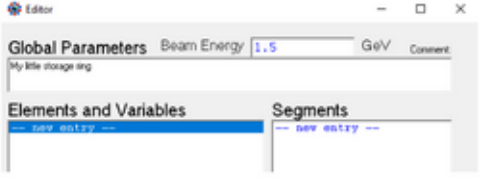
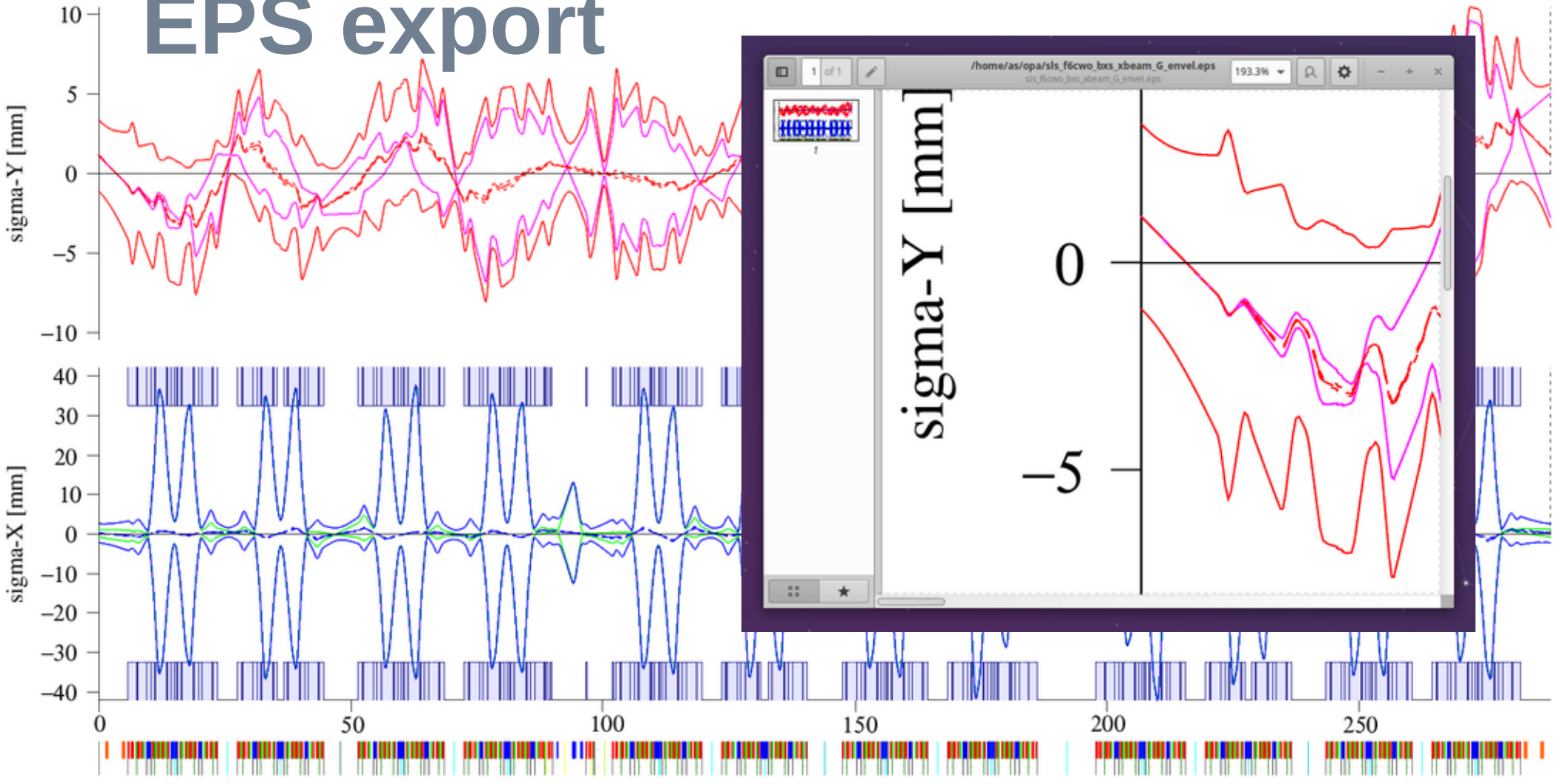


Fig. 2 OPAEditor panel

At first, enter the (maximum) energy of our storage ring, and an optional short description.

EPS export


sigma-Y [mm]
sigma-X [mm]
10
5
0
−5
−10
40
30
20
10
0
−10
−20
−30
−40
0
50
100
150
200
250

/home/as/opa/sls_f6cwo_bxs_xbeam_G_envel.eps
193.3%
sigma-Y [mm]
0
−5

Speed

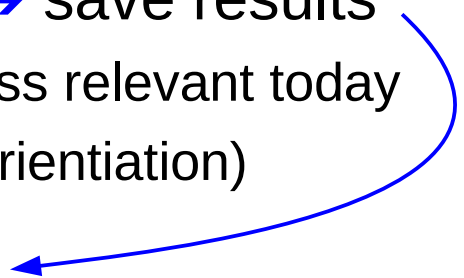
- interactive code : fast response mandatory
- late 1980's PCs: 80286 CPU 10 MHz, 640 KB RAM

➔ memory management

  - temporary allocation, use of pointers

➔ try to split calculations

  - expensive (slow) and infrequent ➔ save results
    - compromise on memory saving – less relevant today
    - compromise on modularity (object orientiation)
  - cheap (fast) and frequent ← use



Example: non-linear driving terms

- 8 1st order Hamiltonian modes e.g. SLS 2.0

$$h_{jklmp} = \sum_{n=1}^{S\cdot K} (b_3 l)_n \, \beta_{xn}^{(j+k)/2} \beta_{yn}^{(l+m)/2} D_{xn}^{p} e^{i[(j-k)\mu_{xn} + (l-m)\mu_{yn}]}$$

  - S = number of sextupoles in lattice 288
  - K = number of kicks per sextupole 3
  - F = number of sextupole families 12

summation over 8 $S\cdot K$ ~ 7000 terms

- Sextupole strengths $\{b_{3n}\}$ are optimizer knobs
  - calculate optical functions only once at beginning:

$$h_{jklmp} = \sum_{q=1}^{F} (b_3 l)_q \left[\sum_{n \in \mathcal{F}_q} \beta_{xn}^{(j+k)/2} \beta_{yn}^{(l+m)/2} D_{xn}^{p} e^{i[(j-k)\mu_{xn} + (l-m)\mu_{yn}]} \right]$$

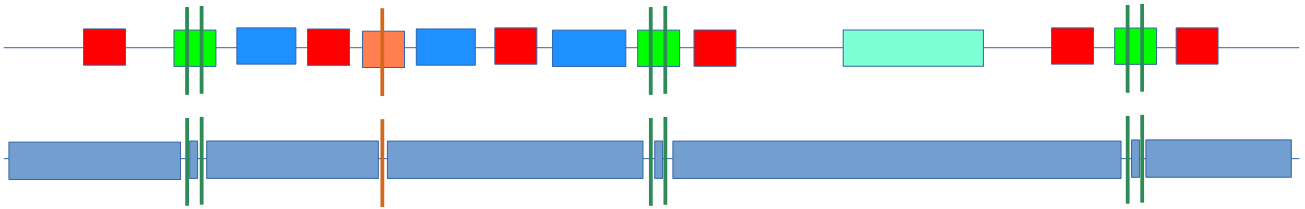
  - save 8×12 matrix → summation over 96 terms only

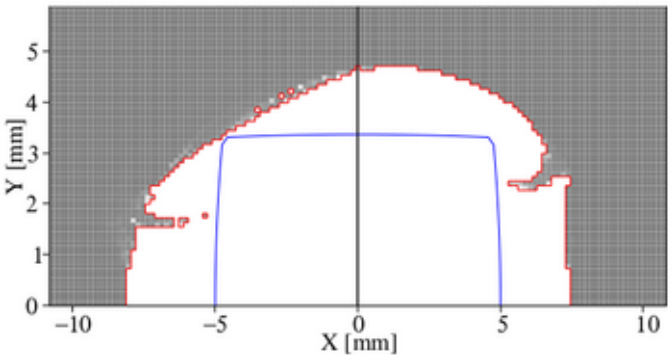
- 11 2nd order modes: replace sum over 11($S\cdot K$)^2 ≈ 1E7 terms by 11×144 matrix

CrX lin	0.00	0.81
CrY lin	0.00	1.57
Qx	H21000	29.22
3Qx	H30000	3.52
Qx	H10110	18.04
Qx-2Qy	H10020	8.63
Qx+2Qy	H10200	29.76
2Qx	H20001	2.40
2Qy	H00201	3.03
Qx	H10002	0.01
	0.00	9.63
	0.00	28.19
	0.00	-24253.57
dQxy,yx	0.00	-14401.80
dQyy	0.00	11829.87
2Qx	H31000	9100.92
4Qx	H40000	11157.37
2Qx	H20110	4259.21
2Qy	H11200	5995.50
2Qx-2Qy	H20020	3211.44
2Qx+2Qy	H20200	2040.90
2Qy	H00310	454.24
4Qy	H00400	459.79
CrX cub	0.00	-45.14
CrY cub	0.00	243.39

Example: tracking

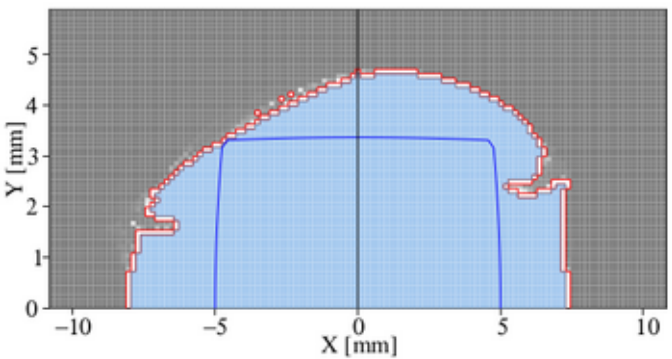
- Concatenation of linear elements
  - Tracking = series of (matrix + kick)





(a) Grid probing, $(dp/p) = 0\%$

- Flood fill (FF) applied to dynamic aperture
  - *Bernard Riemann's idea*
  - factor ~10 speed up compared to grid probing
  - extended to Fast Touschek Tracking (FTT)
    - FF calculation of 3D x,x',Δp acceptance at end of lattice
    - tracking of particles from scattering location to end of lattice to check acceptance



(c) Flood-fill, $(dp/p) = 0\%$

M. Aiba, J. Kallestrup, B. Riemann and AS, Phys. Rev. Accel. Beams 27, 094002 (2024)

```
~/lazarus/opa4$ wc -l *.pas ../com/*.pas
    640 Bucket.pas
    416 CHamLine.pas
     79 ChromGUILib1.pas
     93 ChromGUILib2.pas
   1813 chromlib.pas
    454 CSexLine.pas
    144 EdElCreate.pas
    861 EdElSet.pas
    444 EdSgSet.pas
    421 knobframe.pas
    471 LGBeditorLib.pas
    702 LGBeditor.pas
   2075 mathlib.pas
    480 MomentumLib.pas
   1413 OPAChroma.pas
    145 OPAChromaSVector.pas
    303 OPACurrents.pas
    346 OPAEditor.pas
   2387 OPAElements.pas
   2010 OPAGeometry.pas
   2986 OPAglobal.pas
   1839 opalatticefiles.pas
    901 opamenu.pas
    949 OPAmomentum.pas
   2353 OPAorbit.pas
    472 opatest.pas
   1666 OPAtrackDA.pas
   1422 OPAtrackP.pas
   1726 OPAtrackT.pas
    634 OPAtune.pas
    372 OpticEnvel.pas
   1457 OpticMatch.pas
    119 OpticMatchScan.pas
   2840 opticplot.pas
    615 Opticstart.pas
    251 OpticTune.pas
   1143 opticview.pas
    125 OpticWOMK.pas
    190 texteditor.pas
   1502 tracklib.pas
    349 ../com/asaux.pas
    314 ../com/asfigure.pas
    332 ../com/conrect.pas
   1276 ../com/Vgraph.pas
  41530 total
```

Source code

- 44 Pascal modules `*.pas`
  - **Physics** mainly (not only...) in **`4 units`**
  - own libraries
    - mathematics
    - floating point graphics with eps-export
  - >40.000 lines of Pascal code
- + 28 corresponding Lazarus GUI forms `*.lfm`
  - *including 4 frames* (= self-made GUI components)
- + miscellanous: button icons etc.
- no linked libraries, all code included
  - Numerical Recipes: FFT, SVD, Powell etc.
  - Contour Plot
  - Arithmetic evaluator

Lazarus download | SourceForge.net — Mozilla Firefox

Lazarus download | SourceFo ×

https://sourceforge.net/projects/lazarus/

SOURCEFORGE

For Vendors | Help | Create | Join | Login

Business Software | Open Source Software | SourceForge Podcast | Resources

Search for software or solutic

Home / Open Source Software / Software Development / Integrated Development Environments (IDE) / Lazarus

Lazarus

Rapid applications development tool and libraries for FPC

Brought to you by: mgaertner, mhess, user4martin, vlx, vsnijders

OPEN SOURCE EXCELLENCE SOURCEFORGE

★★★★★ 153 Reviews

Downloads: 17,703 This Week

Last Update: 5 hours ago

Download | Share This

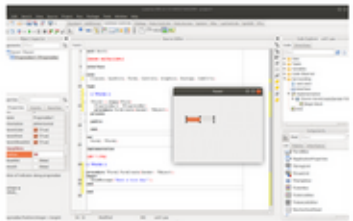
Get an email when there's a new version of Lazarus | Enter your email address | Next

Linux | BSD | Mac | Windows

Summary | Files | Reviews | Support | News | External Link ▾ | Mailing Lists | Code (GIT)

The Lazarus IDE is a stable and feature rich visual programming environment for the FreePascal Compiler. It supports the creation of self-standing graphical and console applications and runs on Linux, FreeBSD, MacOSX and Windows.

Project Samples



Lazarus Free Pascal:

- well established RAD environment
- Cross-platform, open source
- active user community, helpful forum

Recommended Projects

Free Pascal Compiler
Free 32/64/16-bit multi-platform Pascal and Object...

Lazarus Code and Component Repository
Lazarus is a free and open source development tool for...

ZeosLib
Database components for Delphi, FreePascal/Lazarus,...

ACRA
Application crash reports for Android

Top Searches

lazarus

OPA's future ?

- **maintain & consolidate**
  - continue in Lazarus Free Pascal
  - programmers should be familiar with accelerator physics
- **reincarnate**
  - rewrite in modern language (Python):
    - modern well-established language with rich physics and graphics libraries
  - employ young motivated software engineers ☺
  - requires detailed specification of functionality from accelerator physics
- **cannibalize**
  - extract useful algorithms and visualizations and include in other open codes like AT
    - (a previous attempt to link TRACY-2 libraries as shared object to an OPA-like IDL-GUI started promising but failed due to memory conflicts.)