Zgoubi
Recent Developments and Future Plans

Dan T. Abell, Paul Moeller, Boaz Nash, Rob Nagler — RadiaSoft LLC, Boulder, CO
François Méot — BNL C-AD, Upton, NY
Damian Rouson — Sourcery Institute, Oakland, CA
Izaak Beekman — Paratools, Inc, Eugene, OR

Supported in part by the US Department of Energy, Office of Science,
Office of Nuclear Physics, including Award No. DE-SC0017181.

ICAP'18, Key West, FL
23 October 2018

Future Plans for Zgoubi Include Enhancements to both Performance and Usability

Implement a Zgoubi graphical interface in Sirepo.

Implement single-click execution of Zgoubi on available linux clusters.

GUI

Update the Zgoubi code base to the Fortran 2018.

Re-implement Zgoubi's particle update algorithm.

Parallelize Zgoubi using Fortran 2018, including coarrays.

performance

Implement symplectic tracking for field maps.

Add closed-orbit correction to Zgoubi.

capabilities

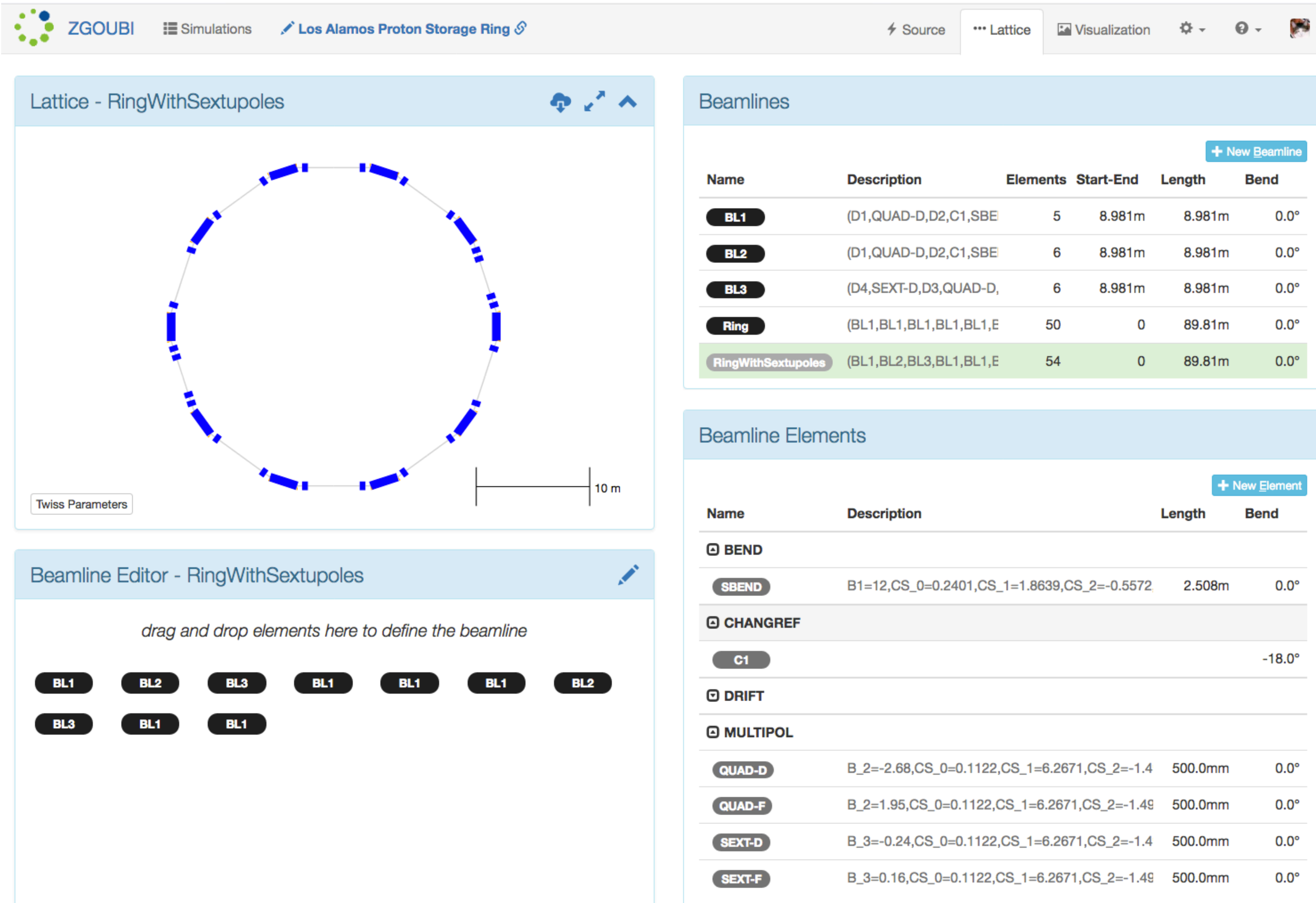
Assess and improve the spin dynamics in electron and ion rings for the eRHIC design.

Benchmark Zgoubi with BMad and other codes used for simulating JLEIC ring designs.

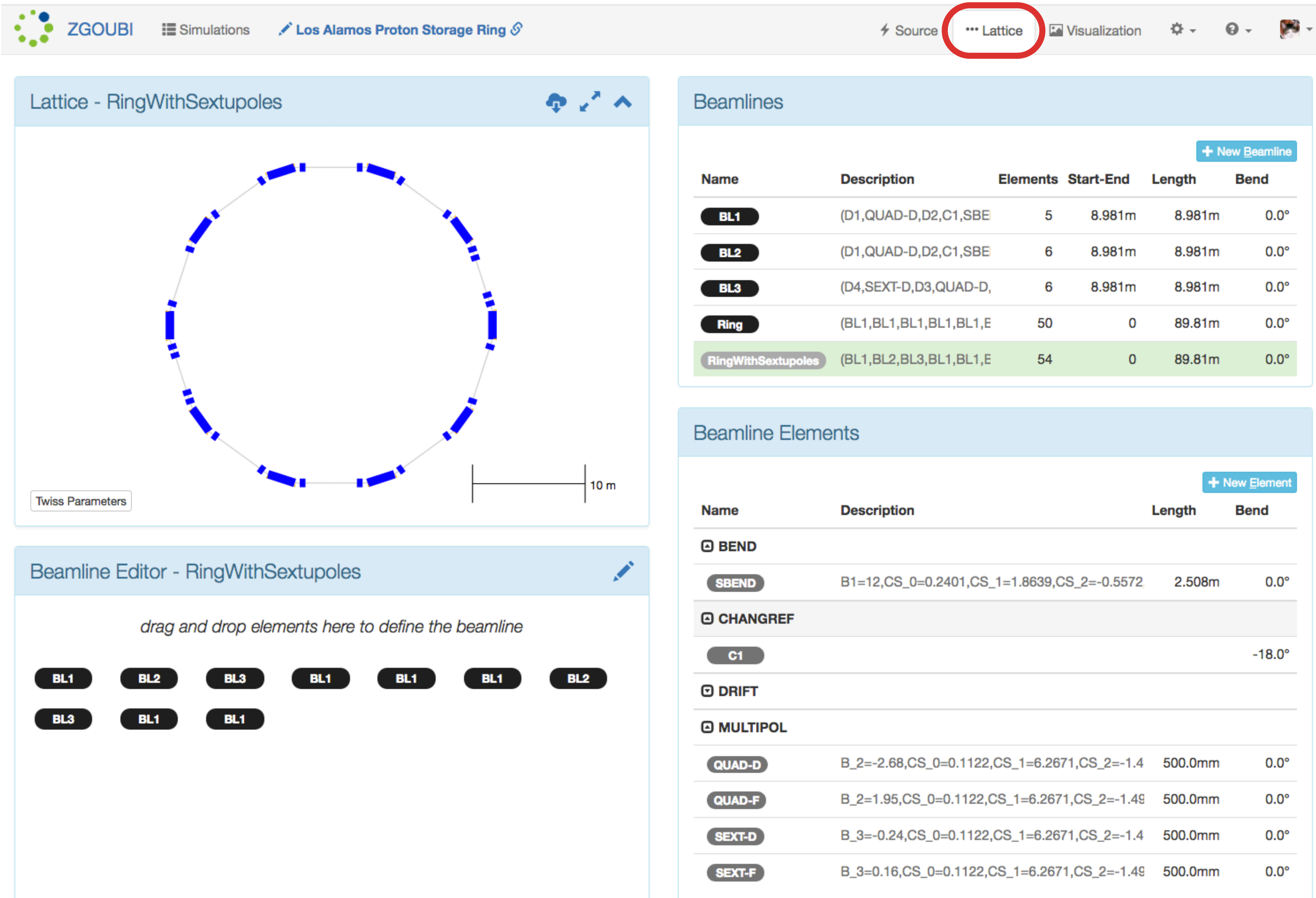
science

Streamline Zgoubi Simulations: Implement a Sirepo Interface for Zgoubi

Sirepo

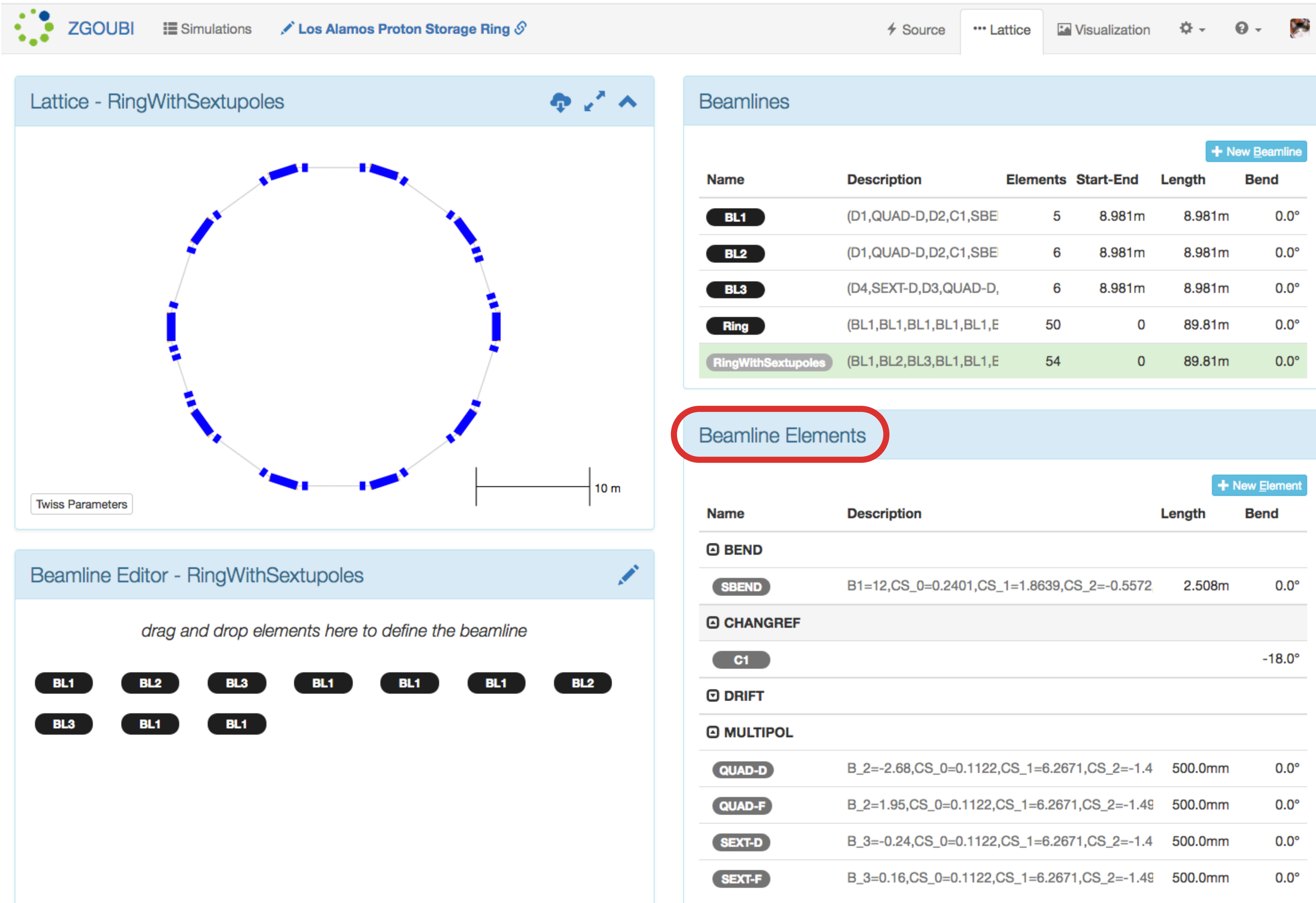


Streamline Zgoubi Simulations: Implement a Sirepo Interface for Zgoubi—cont.



Streamline Zgoubi Simulations: Implement a Sirepo Interface for Zgoubi—cont.

Sirepo

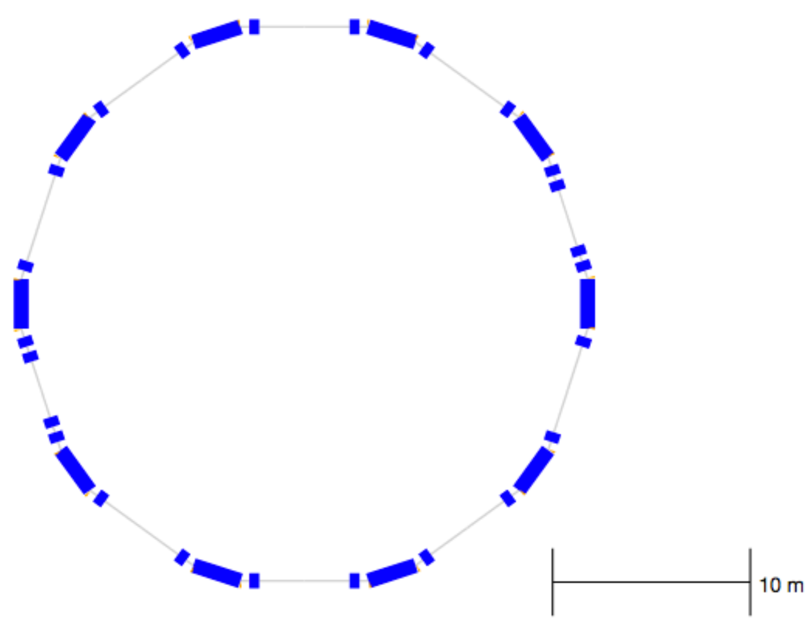


Streamline Zgoubi Simulations: Implement a Sirepo Interface for Zgoubi—cont.

Sirepo

ZGOUBI | Simulations | Los Alamos Proton Storage Ring | Source | Lattice | Visualization

Lattice - RingWithSextupoles



Twiss Parameters

Beamlines

+ New Beamline

Name	Description	Elements	Start-End	Length	Bend
BL1	(D1,QUAD-D,D2,C1,SBE	5	8.981m	8.981m	0.0°
BL2	(D1,QUAD-D,D2,C1,SBE	6	8.981m	8.981m	0.0°
BL3	(D4,SEXT-D,D3,QUAD-D,	6	8.981m	8.981m	0.0°
Ring	(BL1,BL1,BL1,BL1,BL1,E	50	0	89.81m	0.0°
RingWithSextupoles	(BL1,BL2,BL3,BL1,BL1,E	54	0	89.81m	0.0°

Beamline Editor - RingWithSextupoles

drag and drop elements here to define the beamline

BL1 BL2 BL3 BL1 BL1 BL1 BL2 BL3 BL1 BL1

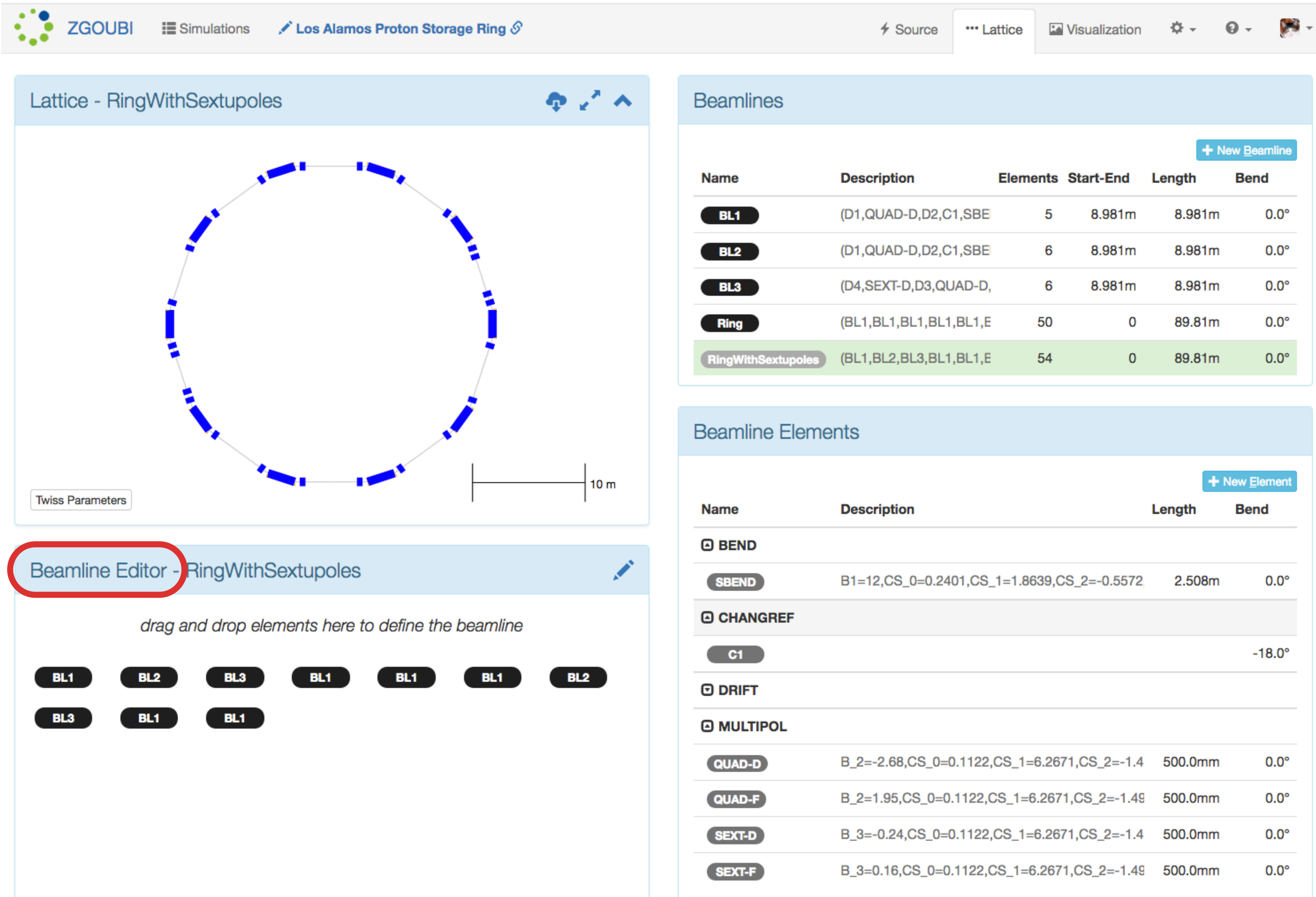
Beamline Elements

+ New Element

Name	Description	Length	Bend
BEND			
SBEND	B1=12,CS_0=0.2401,CS_1=1.8639,CS_2=-0.5572	2.508m	0.0°
CHANGREF			
C1			-18.0°
DRIFT			
MULTIPOL			
QUAD-D	B_2=-2.68,CS_0=0.1122,CS_1=6.2671,CS_2=-1.4	500.0mm	0.0°
QUAD-F	B_2=1.95,CS_0=0.1122,CS_1=6.2671,CS_2=-1.49	500.0mm	0.0°
SEXT-D	B_3=-0.24,CS_0=0.1122,CS_1=6.2671,CS_2=-1.4	500.0mm	0.0°
SEXT-F	B_3=0.16,CS_0=0.1122,CS_1=6.2671,CS_2=-1.49	500.0mm	0.0°

Streamline Zgoubi Simulations: Implement a Sirepo Interface for Zgoubi—cont.

Sirepo

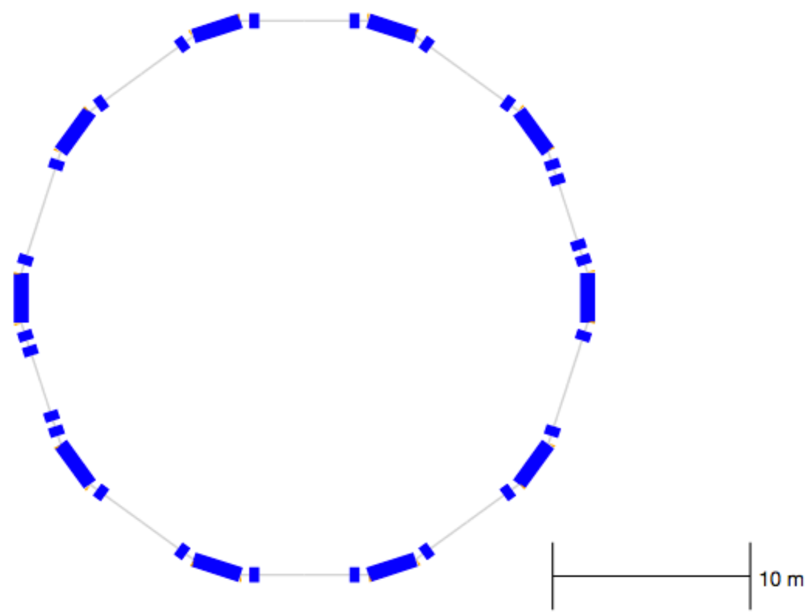


Streamline Zgoubi Simulations: Implement a Sirepo Interface for Zgoubi—cont.

Sirepo

ZGOUBI | Simulations | Los Alamos Proton Storage Ring | Source | Lattice | Visualization

Lattice - RingWithSextupoles



Twiss Parameters

Beamline Editor - RingWithSextupoles

drag and drop elements here to define the beamline

BL1 BL2 BL3 BL1 BL1 BL1 BL2 BL3 BL1 BL1

Beamlines

+ New Beamline

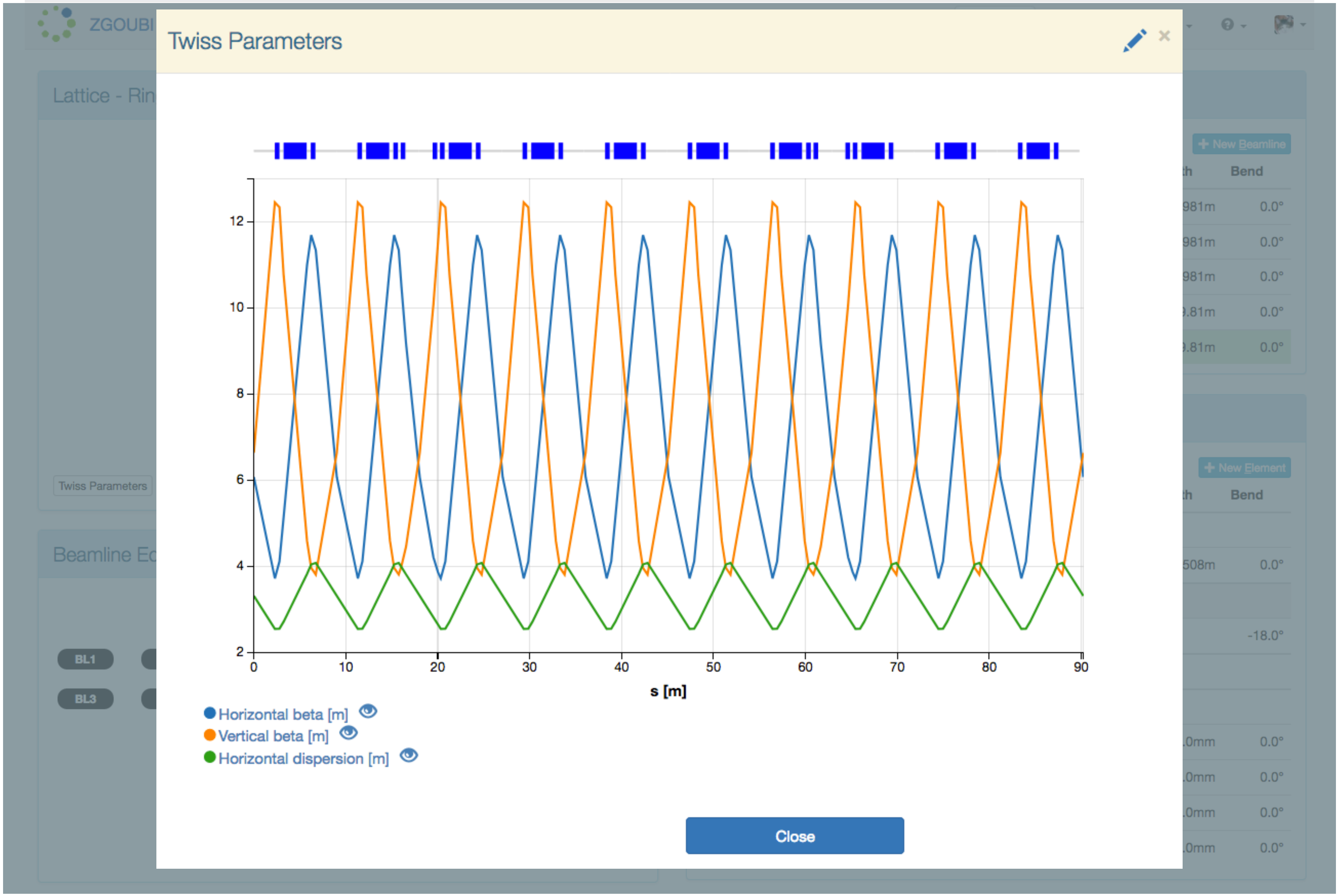
Name	Description	Elements	Start-End	Length	Bend
BL1	(D1,QUAD-D,D2,C1,SBE	5	8.981m	8.981m	0.0°
BL2	(D1,QUAD-D,D2,C1,SBE	6	8.981m	8.981m	0.0°
BL3	(D4,SEXT-D,D3,QUAD-D,	6	8.981m	8.981m	0.0°
Ring	(BL1,BL1,BL1,BL1,BL1,E	50	0	89.81m	0.0°
RingWithSextupoles	(BL1,BL2,BL3,BL1,BL1,E	54	0	89.81m	0.0°

Beamline Elements

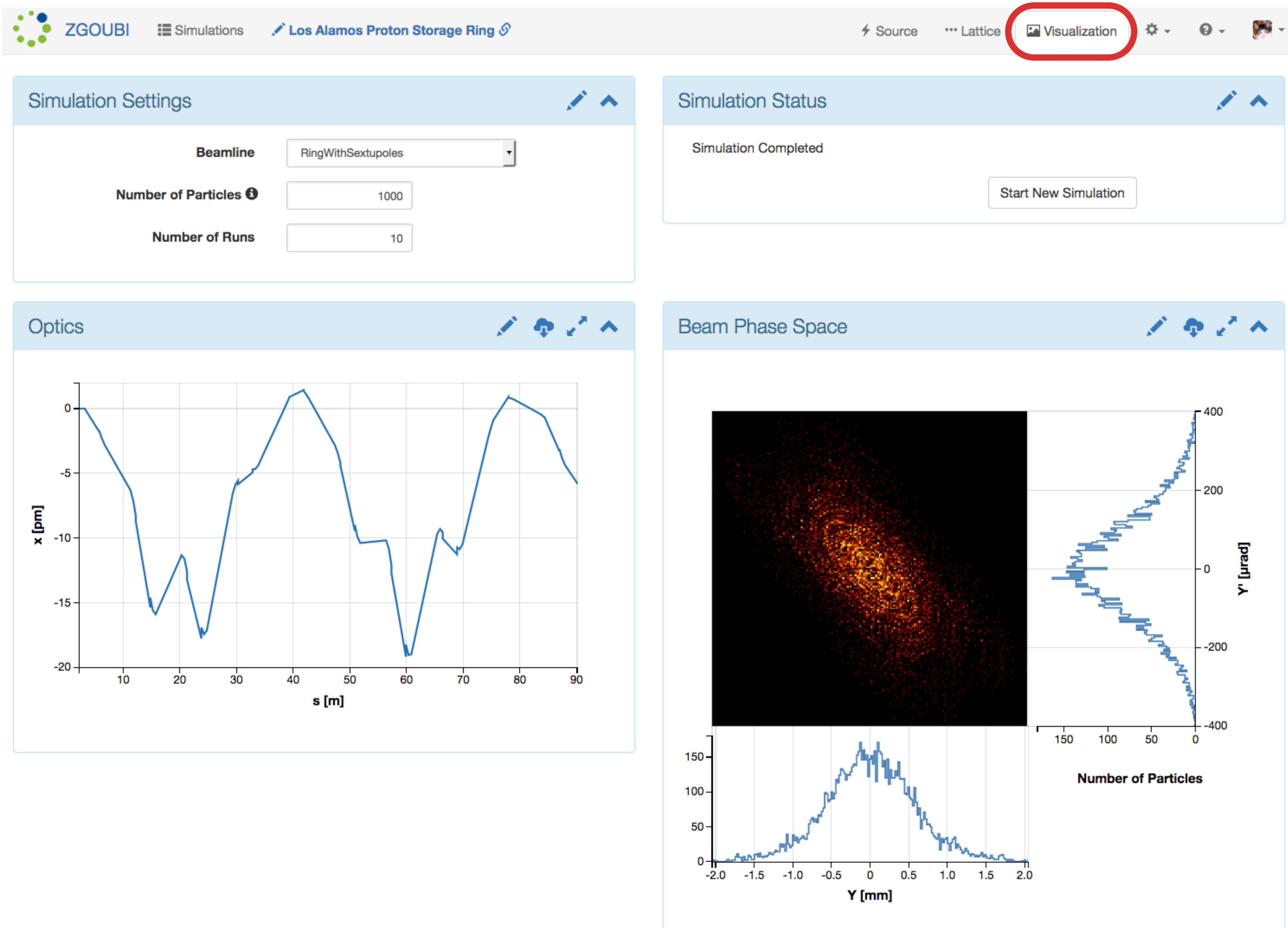
+ New Element

Name	Description	Length	Bend
BEND			
SBEND	B1=12,CS_0=0.2401,CS_1=1.8639,CS_2=-0.5572	2.508m	0.0°
CHANGREF			
C1			-18.0°
DRIFT			
MULTIPOL			
QUAD-D	B_2=-2.68,CS_0=0.1122,CS_1=6.2671,CS_2=-1.4	500.0mm	0.0°
QUAD-F	B_2=1.95,CS_0=0.1122,CS_1=6.2671,CS_2=-1.49	500.0mm	0.0°
SEXT-D	B_3=-0.24,CS_0=0.1122,CS_1=6.2671,CS_2=-1.4	500.0mm	0.0°
SEXT-F	B_3=0.16,CS_0=0.1122,CS_1=6.2671,CS_2=-1.49	500.0mm	0.0°

Streamline Zgoubi Simulations:
Implement a Sirepo Interface for Zgoubi—cont.



Streamline Zgoubi Simulations: Implement a Sirepo Interface for Zgoubi—cont.

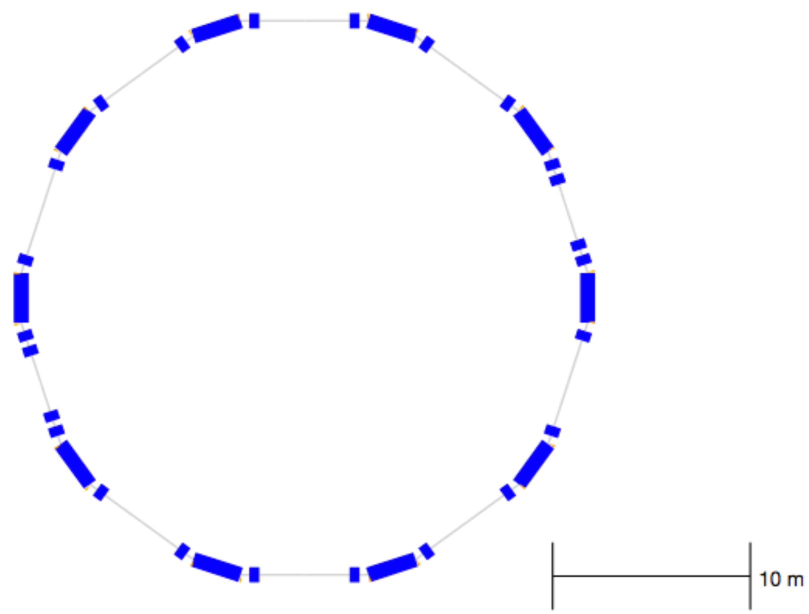


Streamline Zgoubi Simulations: Implement a Sirepo Interface for Zgoubi—cont.

Sirepo

ZGOUBI | Simulations | Los Alamos Proton Storage Ring | Source | Lattice | Visualization

Lattice - RingWithSextupoles



Twiss Parameters

Beamlines

+ New Beamline

Name	Description	Elements	Start-End	Length	Bend
BL1	(D1,QUAD-D,D2,C1,SBE	5	8.981m	8.981m	0.0°
BL2	(D1,QUAD-D,D2,C1,SBE	6	8.981m	8.981m	0.0°
BL3	(D4,SEXT-D,D3,QUAD-D,	6	8.981m	8.981m	0.0°
Ring	(BL1,BL1,BL1,BL1,BL1,E	50	0	89.81m	0.0°
RingWithSextupoles	(BL1,BL2,BL3,BL1,BL1,E	54	0	89.81m	0.0°

Beamline Editor - RingWithSextupoles

drag and drop elements here to define the beamline

BL1 BL2 BL3 BL1 BL1 BL1 BL2 BL3 BL1 BL1

Beamline Elements

+ New Element

Name	Description	Length	Bend
BEND			
SBEND	B1=12,CS_0=0.2401,CS_1=1.8639,CS_2=-0.5572	2.508m	0.0°
CHANGREF			
C1			-18.0°
DRIFT			
MULTIPOL			
QUAD-D	B_2=-2.68,CS_0=0.1122,CS_1=6.2671,CS_2=-1.4	500.0mm	0.0°
QUAD-F	B_2=1.95,CS_0=0.1122,CS_1=6.2671,CS_2=-1.49	500.0mm	0.0°
SEXT-D	B_3=-0.24,CS_0=0.1122,CS_1=6.2671,CS_2=-1.4	500.0mm	0.0°
SEXT-F	B_3=0.16,CS_0=0.1122,CS_1=6.2671,CS_2=-1.49	500.0mm	0.0°

Improve Performance:
Re-implement the Zgoubi Particle Update

Lorentz force law: $\dfrac{\mathrm{d}\vec{p}}{\mathrm{d}t} = q(\vec{E} + \vec{v} \times \vec{B})$

Zgoubi tracks normalized velocity: $\vec{u} = \dfrac{1}{v}\vec{v}$ $\qquad \vec{p} = m\gamma\vec{v} = q(B\rho)\vec{u}$

Then Zgoubi writes the Lorentz force law in the form

$$\frac{\mathrm{d}}{\mathrm{d}s}(B\rho)\vec{u} = (B\rho)'\vec{u} + (B\rho)\vec{u}' = \frac{1}{v}\vec{E} + \vec{u} \times \vec{B}$$

$$\Longrightarrow (B\rho)',\ \vec{u}',\ \ (B\rho)'',\ \vec{u}'',\ \ (B\rho)^{(3)},\ \vec{u}^{(3)},\ \ldots$$

... and the particle update in the form

$$\vec{r}^{f} \approx \vec{r} + \Delta s\, \vec{u} + \frac{\Delta s^2}{2!}\vec{u}' + \cdots + \frac{\Delta s^6}{6!}\vec{u}^{(5)}$$

$$\vec{u}^{f} \approx \vec{u} + \Delta s\, \vec{u}' + \frac{\Delta s^2}{2!}\vec{u}'' + \cdots + \frac{\Delta s^5}{5!}\vec{u}^{(5)}$$

Improve Performance: Re-implement the Zgoubi Particle Update

Lorentz force law: $\dfrac{\mathrm{d}\vec{p}}{\mathrm{d}t} = q(\vec{E} + \vec{v} \times \vec{B})$

Zgoubi tracks normalized velocity: $\vec{u} = \dfrac{1}{v}\vec{v}$ $\qquad \vec{p} = m\gamma\vec{v} = q(B\rho)\vec{u}$

Then Zgoubi writes the Lorentz force law in the form

$$\frac{\mathrm{d}}{\mathrm{d}s}(B\rho)\vec{u} = (B\rho)'\vec{u} + (B\rho)\vec{u}' = \frac{1}{v}\vec{E} + \vec{u} \times \vec{B}$$

distance along trajectory

$$\Longrightarrow (B\rho)',\ \vec{u}',\ \ (B\rho)'',\ \vec{u}'',\ \ (B\rho)^{(3)},\ \vec{u}^{(3)},\ \ldots$$

... and the particle update in the form

$$\vec{r}^{f} \approx \vec{r} + \Delta s\,\vec{u} + \frac{\Delta s^2}{2!}\vec{u}' + \cdots + \frac{\Delta s^6}{6!}\vec{u}^{(5)}$$

$$\vec{u}^{f} \approx \vec{u} + \Delta s\,\vec{u}' + \frac{\Delta s^2}{2!}\vec{u}'' + \cdots + \frac{\Delta s^5}{5!}\vec{u}^{(5)}$$

Improve Performance:
Re-implement the Zgoubi Particle Update—cont.

If *no* electric field, then $\vec{u}' = \vec{u} \times \vec{b}$, where $\vec{b} = \dfrac{1}{B\rho}\vec{B}$.

Successive derivatives of $\vec{u}$ are computed as

$$\vec{u}'' = \vec{u}' \times \vec{b} + \vec{u} \times \vec{b}', \quad \vec{u}''' = \vec{u}'' \times \vec{b} + 2\vec{u}' \times \vec{b}' + \vec{u} \times \vec{b}'', \quad \ldots .$$

Successive derivatives of $\vec{b}$ are computed as

$$\vec{b}' = \sum_i \frac{\partial \vec{b}}{\partial x_i} u_i$$

$$\vec{b}'' = \sum_i \frac{\partial \vec{b}}{\partial x_i} u_i' + \sum_{ij} \frac{\partial^2 \vec{b}}{\partial x_i \partial x_j} u_i u_j$$

$$\vdots$$

Improve Performance: *Re-implement the Zgoubi Particle Update*—cont.

If *no* electric field, then $\vec{u}' = \vec{u} \times \vec{b}$, where $\vec{b} = \dfrac{1}{B\rho}\vec{B}$.

Successive derivatives of $\vec{u}$ are computed as

$$\vec{u}'' = \vec{u}' \times \vec{b} + \vec{u} \times \vec{b}', \quad \vec{u}''' = \vec{u}'' \times \vec{b} + 2\vec{u}' \times \vec{b}' + \vec{u} \times \vec{b}'', \quad \ldots .$$

Successive derivatives of $\vec{b}$ are computed as

$$\vec{b}' = \sum_i \frac{\partial \vec{b}}{\partial x_i} u_i$$

$$\vec{b}'' = \sum_i \frac{\partial \vec{b}}{\partial x_i} u_i' + \sum_{ij} \frac{\partial^2 \vec{b}}{\partial x_i \partial x_j} u_i u_j$$

$$\vdots$$

Truncated power series algebras (TPSAs) perform essentially the same computations; hence we can take advantage of efficient TPSA algorithms. Doing so reduces both memory footprint and corresponding arithmetic by roughly a factor of 3.

Improve Performance: Parallelize Zgoubi using Fortran 2018

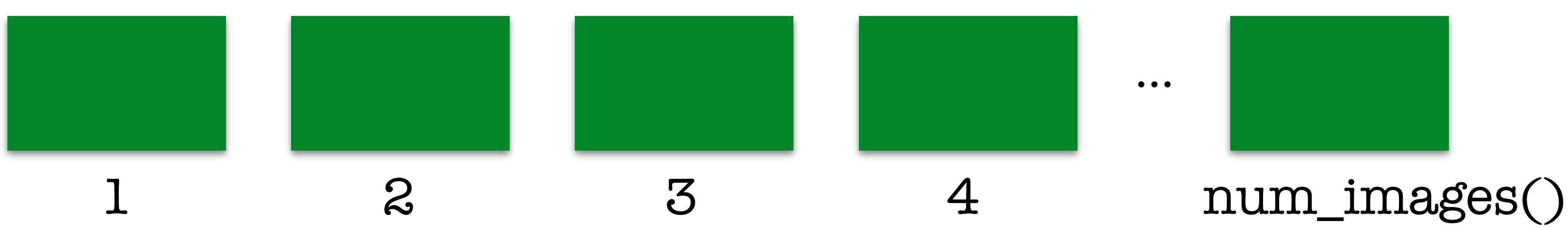
The coarray programming model is designed to answer the question 'What is the smallest change required to convert Fortran into a robust and efficient parallel language?'.

—— John Reid, *ISO/IEC JTC1/SC22/WG5 N1824* (2010)

Coarray syntax implements a Single Program Multiple Data (SPMD) model.

A single program is replicated in units called *images*.

The number of images may be chosen at compile time.




```
real :: a[*]   ! scalar coarray
real, dimension(10) :: x[*], y[*]   ! array coarray
real :: m(0:21,6)[*]   ! matrix coarray
type(particle) :: ptcl(128)[*]   ! derived type coarray
real :: array(10,20)[10,-1:8,0:*]   ! more complex coarray

x(:) = y(:)[q]   ! access remote data on image q
```

Improve Performance: Parallelize Zgoubi using Fortran 2018

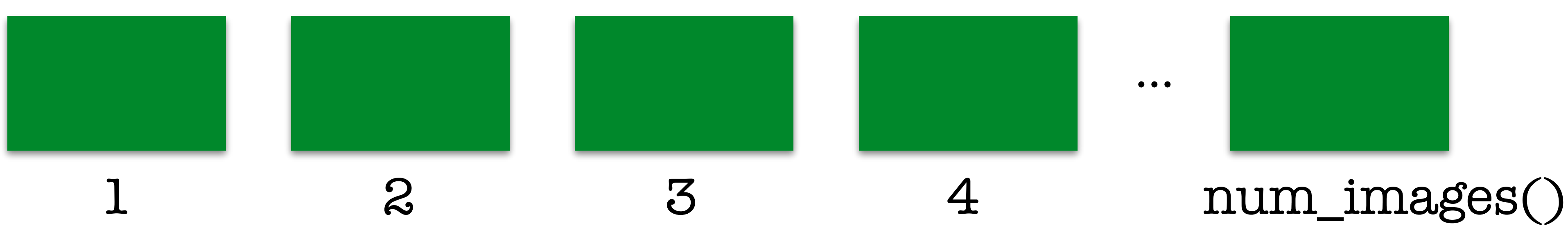
The coarray programming model is designed to answer the question 'What is the smallest change required to convert Fortran into a robust and efficient parallel language?'.

— John Reid, *ISO/IEC JTC1/SC22/WG5 N1824* (2010)

Coarray syntax implements a Single Program Multiple Data (SPMD) model.

A single program is replicated in units called *images*.

The number of images may be chosen at compile time.




```
real :: a[*]  ! scalar coarray
real, dimension(10) :: x[*], y[*]  ! array coarray
real :: m(0:21,6)[*]  ! matrix coarray
type(particle) :: ptcl(128)[*]  ! derived type coarray
real :: array(10,20)[10,-1:8,0:*]  ! more complex coarray

x(:) = y(:)[q]  ! access remote data on image q
```

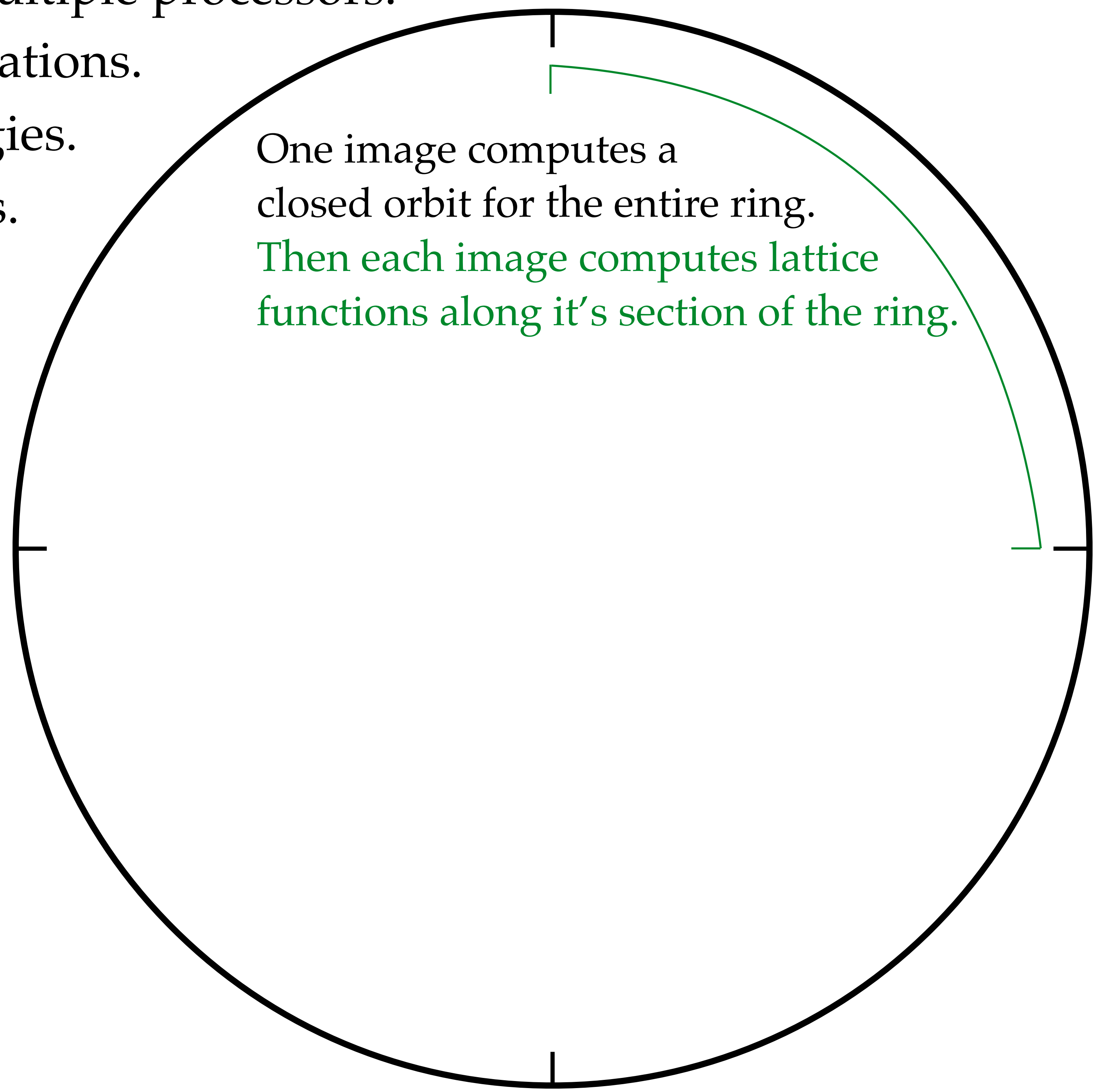
A coindex indicates communication.
The programmer must ensure that
* coarray indices are properly resolved,
* synchronization occurs as appropriate.

The new Fortran standard includes new intrinsics that simplify the process.

Improve Performance: Parallelize Zgoubi using Fortran 2018—cont.

Opportunities for paralellizing Zgoubi:

- Distribute particles on multiple processors.
- Distribute lattice computations.
- Distribute machine energies.
- Distribute misalignments.
- Various scans:
  - dynamic aperture;
  - polarization studies
  - etc.



Thank you!

Supported in part by the US Department of Energy, Office of Science, Office of Nuclear Physics, including Award No. DE-SC0017181.

Sirepo https://sirepo.com/