AUCKLAND BIOENGINEERING INSTITUTE
THE UNIVERSITY OF AUCKLAND
NEW ZEALAND

OpenCMISS-Zinc: Visualising Research

Hugh Sorby, Alan Wu, Richard Christie, The University of Auckland, New Zealand

Visualising Research

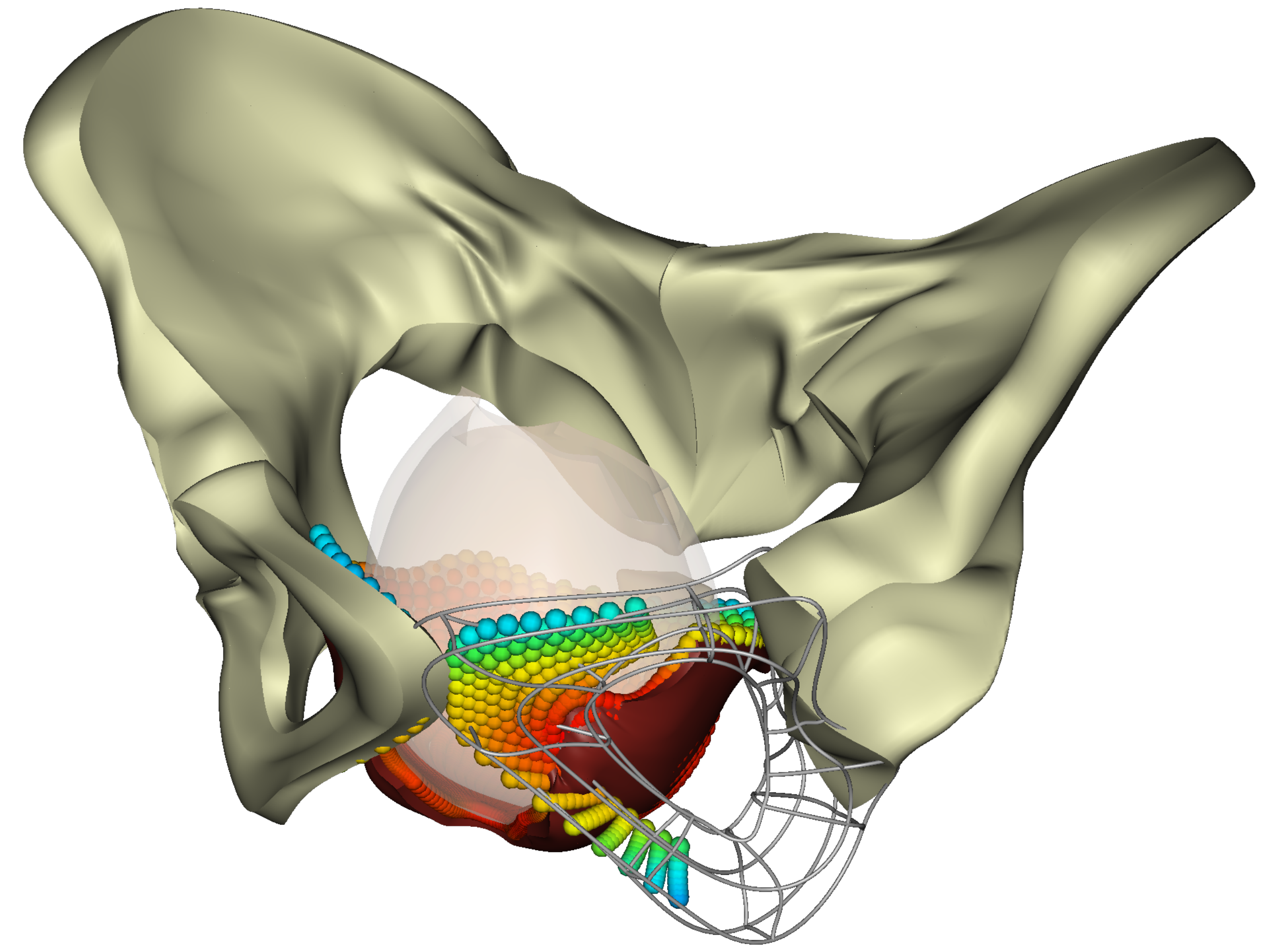
Modelling and imaging are key components of research and applications in Bioengineering, each capable of producing large datasets of high dimensionality. Interactive visualisation is essential for communicating the meaningful information in these datasets to researchers, customers, clinicians and patients.

Beyond Visualisation

OpenCMISS-Zinc ("Zinc") is a software library for creating interactive graphical modelling applications, combining a dynamic model representation including images, high quality graphics generation and rendering, and many features for interacting with and manipulating the model. Target uses for the Zinc library include:

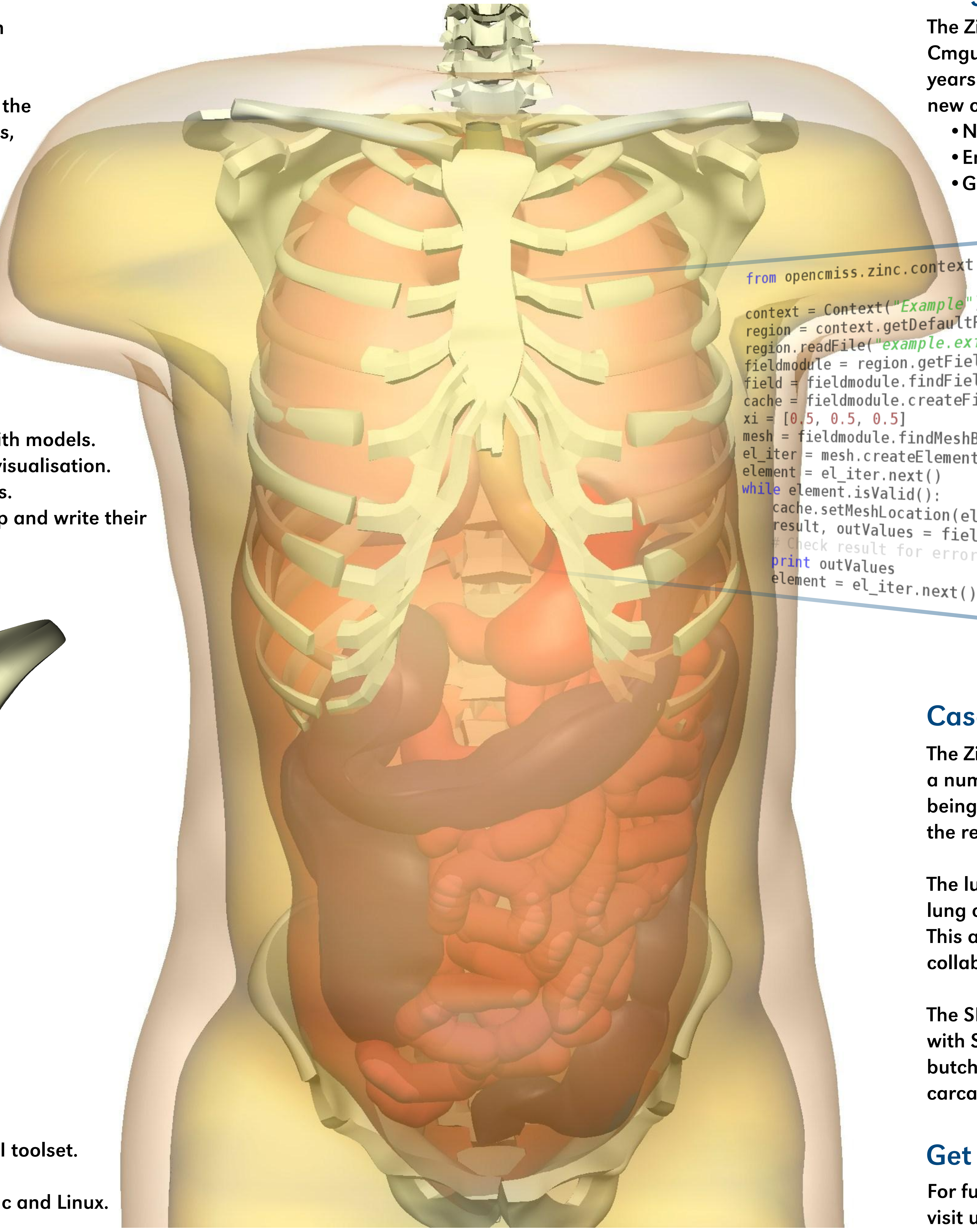
- Clinical applications combining medical imaging with models.
- Custom interfaces for model creation, editing and visualisation.
- General purpose scientific visualisation applications.

Zinc is easy enough for individual researchers to pick up and write their own scripts and tools, with or without graphics.



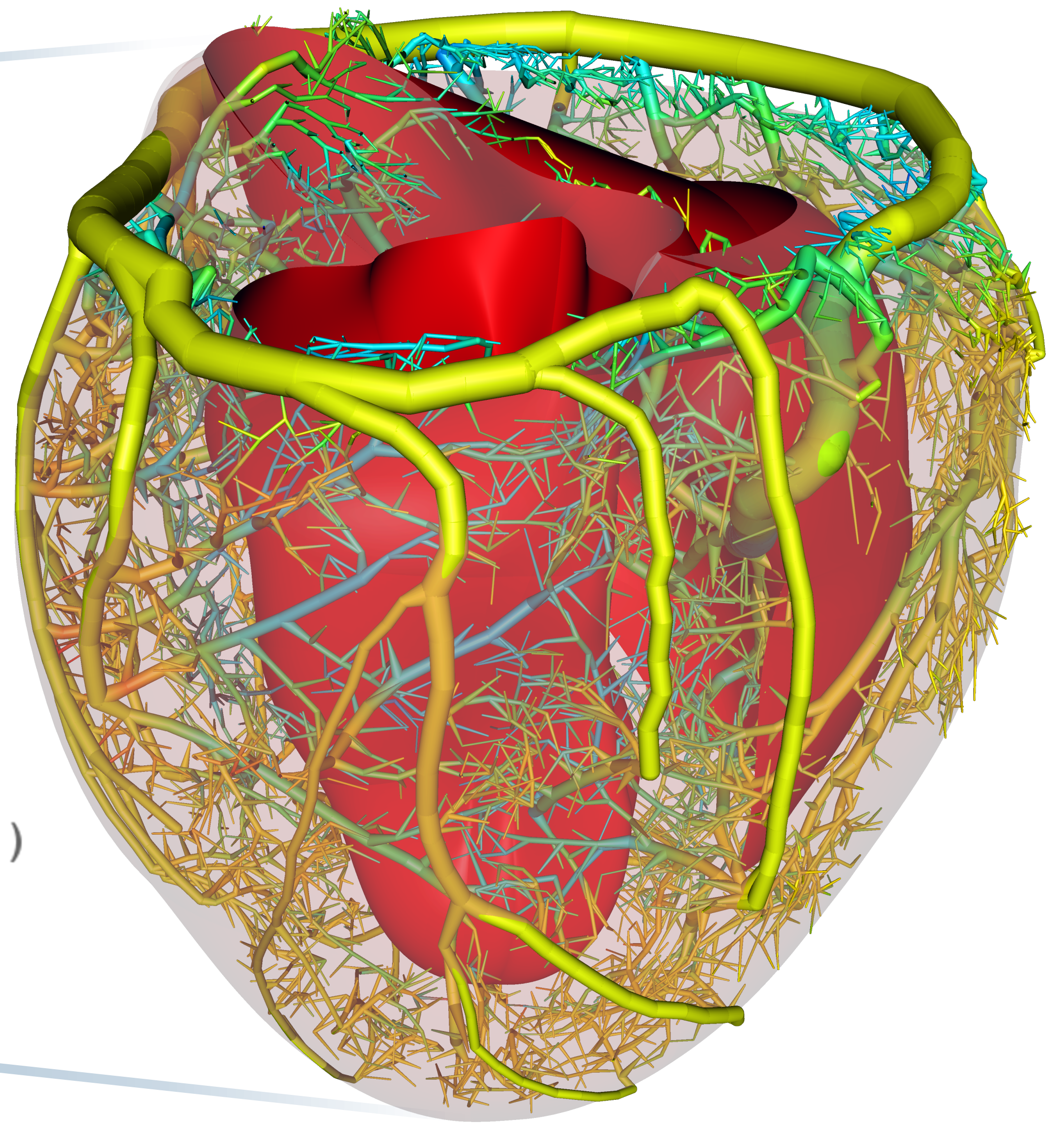
Features of the Zinc library:

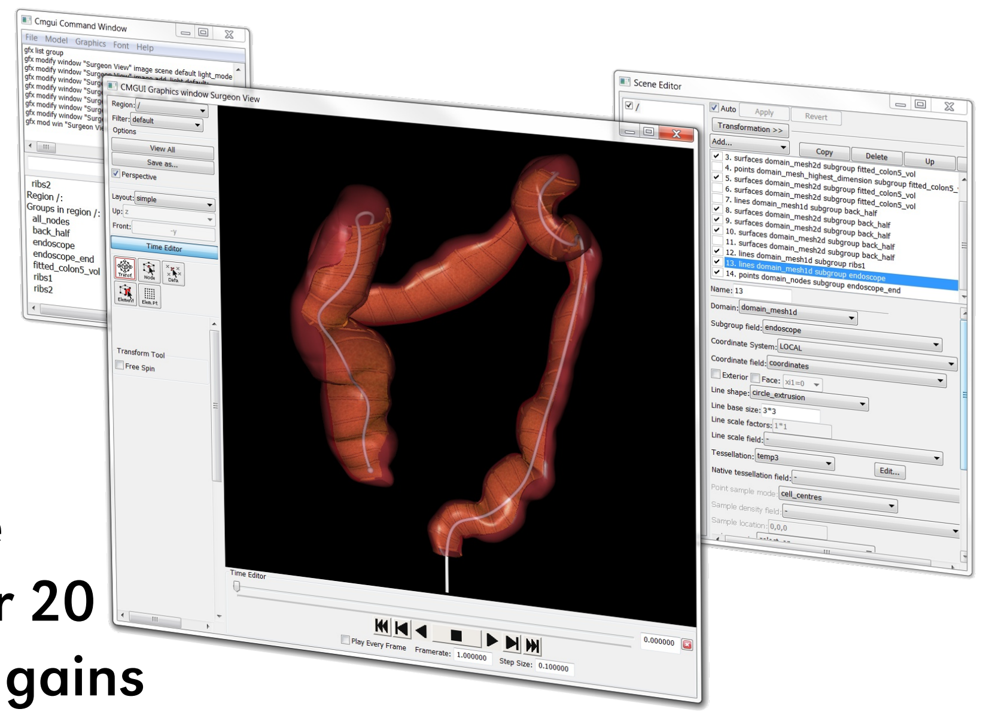
- Rich model representation with finite elements and images including model creation and editing.
- Field abstraction including deriving fields by mathematical operators, image processing.
- Powerful control of graphics generation from fields.
- High quality OpenGL rendering, independent of GUI toolset.
- Interoperates with OpenCMISS-Iron.
- C, C++ and Python APIs, available for Windows, Mac and Linux.
- Actively developed and supported.



```
from opencmiss.zinc.context import Context

context = Context("Example")
region = context.getDefaultRegion()
region.readFile("example.exfile")
fieldmodule = region.getFieldmodule()
field = fieldmodule.findFieldByName("coordinates")
cache = fieldmodule.createFieldcache()
xi = [0.5, 0.5, 0.5]
mesh = fieldmodule.findMeshByDimension(3)
el_iter = mesh.createElementiterator()
element = el_iter.next()
while element.isValid():
    cache.setMeshLocation(element, xi)
    result, outValues = field.evaluateReal(cache, 3)
    # Check result for errors, Use outValues
    print outValues
    element = el_iter.next()
```





Cmgui: Built on Zinc

The Zinc Library is the core engine behind the Cmgui visualisation application, sharing over 20 years development. As Zinc develops, Cmgui gains new capabilities including:

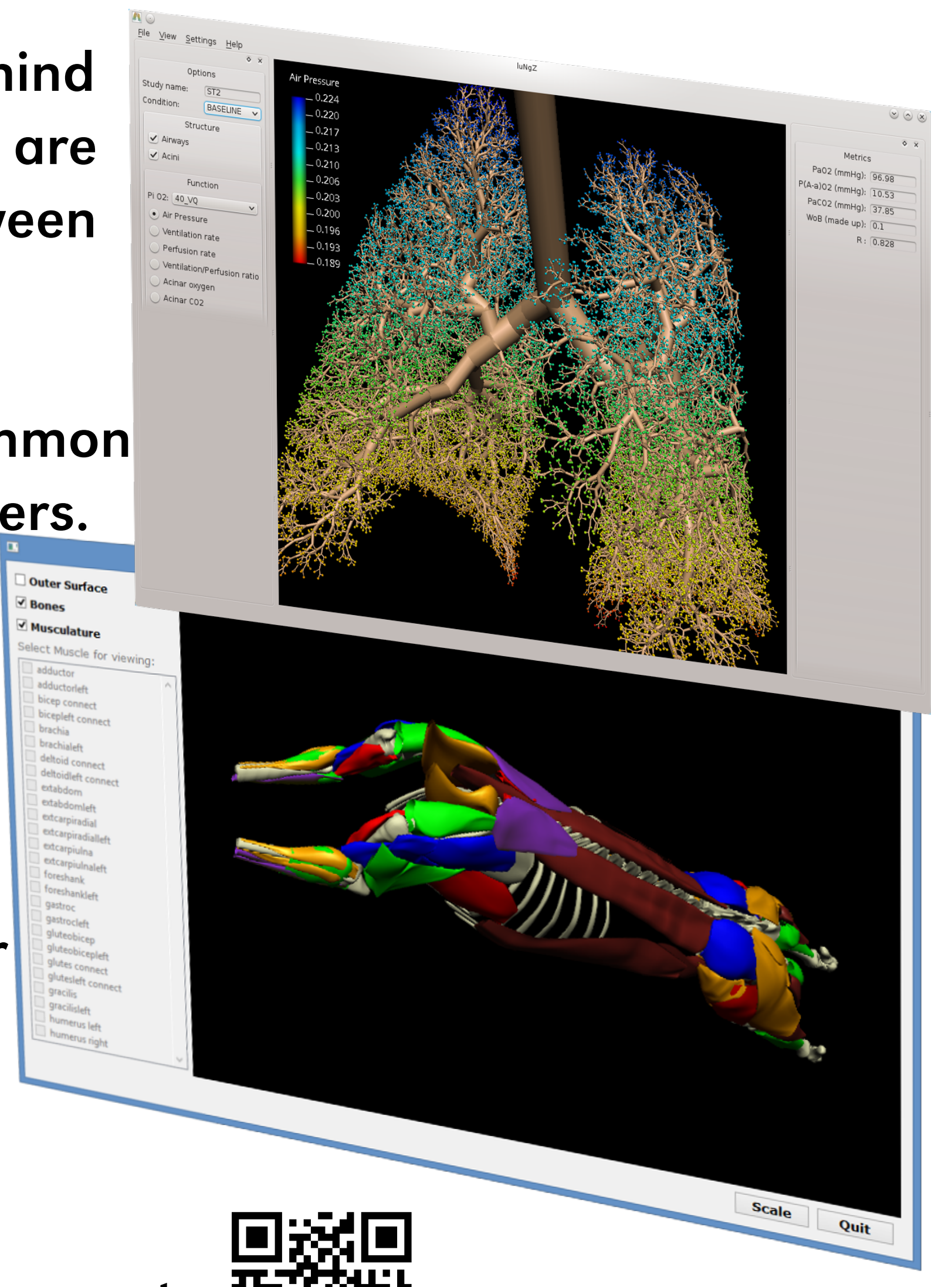
- New file format support.
- Enhanced visualisation and interaction features.
- Greater stability.

Case Studies:

The Zinc library is the visualisation engine behind a number of applications. These applications are being used to bridge the knowledge gap between the researcher and the target audience.

The luNgZ application shows the effect of common lung conditions on a number of lung parameters. This application is used in a research collaboration with F&P Healthcare.

The SFFLCM application is being developed with Silver Fern Farms to inform the abattoir butcher about the optimal cut for a particular carcass.



Get Started Using Zinc:

For further information, documentation and support visit us at http://www.opencmiss.org/zinc