Spatial Reference Model (SRM)

An Overview

http://www.sedris.org/srm.htm

6 January 2004

Lake Buena Vista, Florida

Dr. Paul Berner
SEDRIS Core Team
ISO/IEC 18026 Editor
Berner@Consultant.com

Dr. Ralph M. Toms
SRI International
ISO/IEC 18026 Editor
ralph_toms@sri.com

SRM tutorial

DESCRIPTION:
The SRM is a unified approach for the representation and use of spatial location information. Designed as a standalone technology, the SRM is a unified approach for the representation and use of spatial location information. Designed as a standalone technology, the SRM provides a complete and concise treatment of the different descriptions of spatial location, and precisely defines the relationship between various spatial reference frames. This tutorial provides a detailed review of the SRM framework and concepts with an emphasis on the modeling implications of different ways of representing spatial location. Challenges and issues in implementing precise and efficient coordinate operations are also discussed.

WHO SHOULD ATTEND:
Those interested in gaining a more complete understanding of the SRM and the theory of accurately representing spatial locations in modeling.

PREREQUISITE:
Prior knowledge of other SEDRIS technologies is not required, however prior attendance at either the "Introduction to SEDRIS for Managers" or "SEDRIS - The Technology Components" tutorial is recommended.

WHAT TO EXPECT:
At completion, the attendee has an appreciation for the complexities involved in accurately representing spatial location. Topics such as object reference models, spatial reference frames, ellipsoids and geoids are covered, among others.

Prerequisite

- To get the most from this tutorial, we assume you know the following, as a prerequisite to this session:
  - An understanding of basic mathematical concepts
  - An interest in coordinate representation, conversion, and transformation techniques
  - Familiarity with the SEDRIS technology components and how they fit together. We assume you have attended:
    - ***"Introduction to SEDRIS for Managers"*** **or**
    - ***"SEDRIS - The Technology Components"***

Outline

- Introduction
  - scope
  - goals
- SRM Concepts
  - key concept preview
  - object-space and position-space
  - reference datums and object reference models
  - coordinate systems
  - spatial reference frames
  - vertical offset surfaces
  - spatial operations and the API
- Computational considerations
- Spatial operation validity, conformance, accuracy
- Structure of the SRM ISO standard

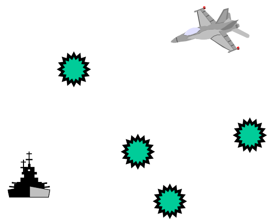
The Spatial Reference Model (SRM) is a general concept

There are two aspects of the SRM discussed in this tutorial

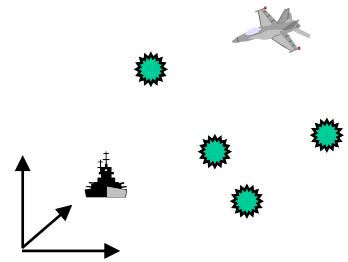
- One aspect is the SRM as an International Standard.
  - ISO/IEC 18026 Spatial Reference Model.
  - Under development, the final committee draft is to be released summer ‘04.
  - This tutorial uses the nomenclature, definitions and concepts of ISO/IEC 18026.
  - Consistency with other International Standards is promoted.
  - Efficiency of algorithms or their implementation is not part of the ISO/IEC 18026 specification.

- Another aspect is the implementation of the SRM in SEDRIS
  - The SEDRIS implementation is designed to be compliant with the emerging 18026 International Standard.
  - The SEDRIS implementation is an application driven development.
  - A software architecture and spatial operations API is required.
  - Algorithms for spatial operations in spatial frames are designed to be accurate robust and efficient.
  - The SEDRIS software has been subject to extensive testing.

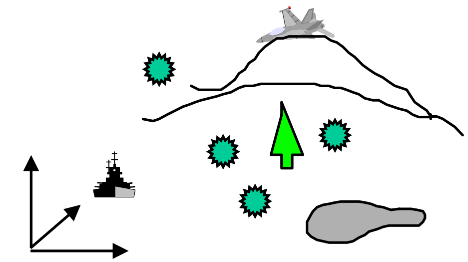
In the beginning ...



Systems



Systems, where?



Systems, and what else?

The void ...

The environment starts with locating your systems; sometimes that's about all you could afford in legacy simulations.

The environment continues with defining the context within which systems *interact*; and that context can advantage, or disadvantage, them ...

- Defining and using a consistent spatial reference framework is critical for simulation *interoperability*
  – System models (men, equipment, material, …)
  – Environmental: data, models, phenomena
- At all levels of simulation detail/resolution
  – Aggregated / Entity; Constructive / Virtual / Live

Why is a Spatial Reference Model (SRM) needed?

- The Modeling & Simulation (M&S) community has not been consistent in the treatment of models of the Earth and related coordinate systems.
- Consistency is required for joint distributed simulation in order to:
  - Achieve a reasonably level playing field,
  - To support meaningful VV&A.
- A number of different Earth reference models (ERMs) are currently employed and this affects:
  - Representation of the environment in M&S and authoritative data bases.
  - Dynamics formulations, both kinematics and kinetics (movement).
  - Acquisition modeling and processing (inter-visibility).
- Loss-less and accurate spatial operations are critical to representation and interchange of location data.
- A nomenclature inconsistency exists.
  - For example, how do these variables relate?

    Altitude, elevation, height, geodetic height, ellipsoidal height, orthometric height, height above sea level, height above mean sea level, terrain height, pressure altitude, temperature altitude, nap of the Earth, ...

Other interoperability examples

- It isn't just simulations that have potential interface problems due to spatial referencing.
- Real (embedded) systems also need to communicate spatially referenced data.
- Test range operation is another application domain where consistent spatial referencing is important, particularly if multiple ranges share real-time information.

The federation of live, virtual and constructive systems requires a standardized SRM.

Why is accuracy needed for coordinate transformations?

- In the past, it has been difficult to measure positions on the surface of the Earth to better than 1 metre.
  - This situation is changing due to new technology developments.
  - GPS now can achieve absolute accuracy of about 21 cm (SEP 90%) over large regions.
- Real-world applications mostly use relative coordinate systems
  - Dynamically correct location errors by using on-board sensors.
- In the simulation environment, both absolute and relative spatial referencing must be accurately portrayed.
- Mixing of live & synthetic environments has special accuracy requirements*.
- Mission planning, rehearsal, & conduct of real operations have situation-dependent accuracy requirements.

* ***Lucha, G. V., On the Consequences of Neglecting Measurement Accuracy Issues in Live and Virtual interactions", SIW Spring 1997***

Target shooting at a visible target in the real world

(Relative coordinate system)

- Set up paper target with aim-point approximately 1000 metres away.
- Shoot N rounds.
- Measure miss distances using bullet holes on paper target, and compute CEP or some other measure of accuracy.

Never used any precise location or environmental data.

Simulation of target shooting at a visible target

(Simulation of a relative coordinate system)

- Select lococentric Euclidean 3D CS, origin at the shooter.
- Define coordinates of the target & shooter.
  - **Both with target plane oriented perpendicular to line of sight.**
- Develop an aiming model with random inputs.
- Define shoot time, T.
- Integrate bullet trajectory in time, from T until it pierces the plane of the target (need air temperature, density, speed of sound, wind, *etc*.)
  - **May have to access geodetic system for each of the environmental parameters.**
  - **Will need an iterative scheme to get the impact point.**
- Compute radial miss at target plane impact.

Any errors made in any of the position-location computations, including those needed to compute the correct environmental parameters, can and will dilute the accuracy of the result.

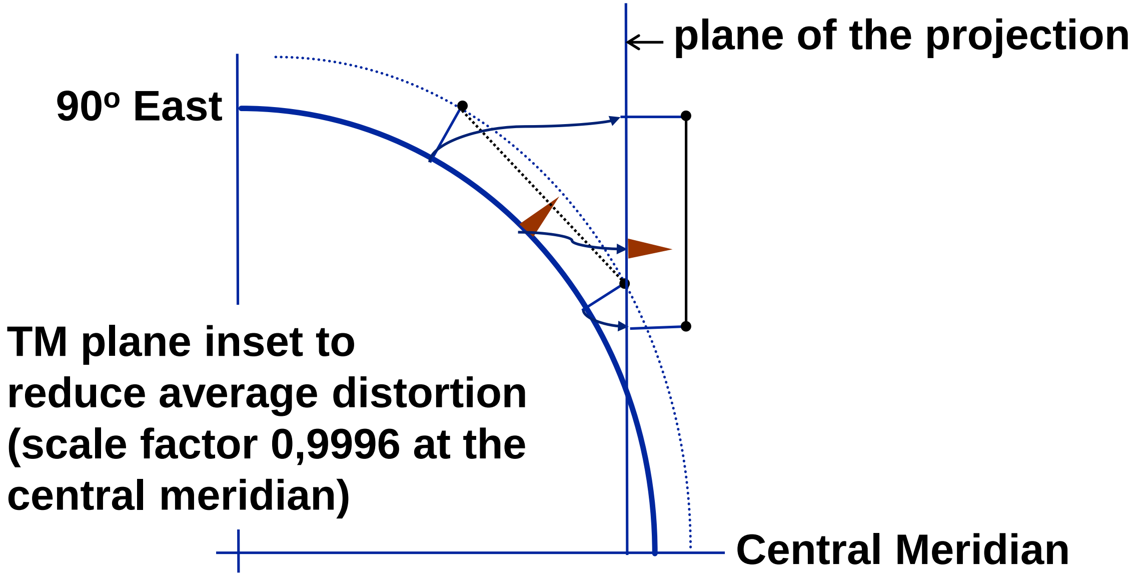
Servicing a non-visible target in the real world

- The problem: An artillery piece shoots at a target coordinates provided from an external source.
- The shooter has its location in some coordinate system, e. g. Augmented TM.
- The target coordinates may be known to some accuracy in geodetic latitude, longitude & geodetic height.
- The fire control system may or may not require conversions to another coordinate system, for example a lococentric Euclidean 3D coordinate system.
- The fire control system computes firing angles based on its internal mechanization and may model wind, air density and other environmental parameters.
- The round impact and miss distances are observed by a forward observer who feeds information back to the firing site.
- The process is completed when the impact error is small enough.

Use location data but no precisely measured environmental data.
Simulation of this scenario requires detailed dynamics and environmental models

Augmented Transverse Mercator(ATM) example

Several distortions are introduced, especially at the higher latitudes.



The results of such elevation angle and range distortions may not be so apparent when all simulations involved use ATM. However, in a federation involving real world coordinate systems the distortions may become evident. Use of ATM increases visibility, causes interactions to prosecute too fast, leads to an uneven playing field and is not recommended for use in joint simulations.

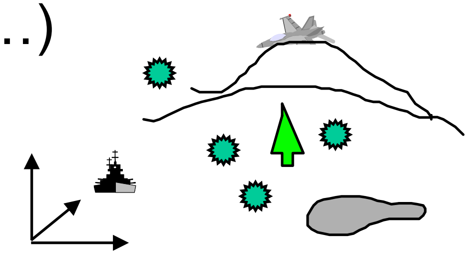
SRM requirements

- *Completeness*, must:
  - include coordinate systems in wide usage.
  - tie those systems together into a commonly used framework.
  - educate the system developer.
    - *E.g.*, What's a horizontal datum, a vertical datum, a geoid?
- *Accuracy*
  - Generally higher than required for C4ISR systems.
  - E.g., Typically better than 1 cm. up past geosynchronous orbit (nominally 1 mm near the ERM surface).
  - Computation of operations in spatial frames must not impact application validity.
- *Performance*
  - Never fast enough!
  - Many environmental data sets dominated by location data.
    - Therefore efficient inter-conversion is key to meeting 72 hour "ready-to-run" mandate.
  - Federate costs for distributed simulation using heterogeneous coordinate systems can be substantial (*e.g.*, 20% or more).

SEDRIS algorithms and implementations are very accurate and 2-5x faster than algorithms/implementations reported in the literature.

Location is not enough

- The key in Modeling and Simulation is not so much where things are, but *what they can interact with* …
  - Remembering that "what" includes both systems and components of the environment itself
- A complete SRM must address:
  - Location
  - Distance (range, line of sight, geodesics…)
  - Vector quantities, such as direction (azimuth and elevation angle)
- *Location interconversions between common spatial reference frames do not necessarily preserve these.*



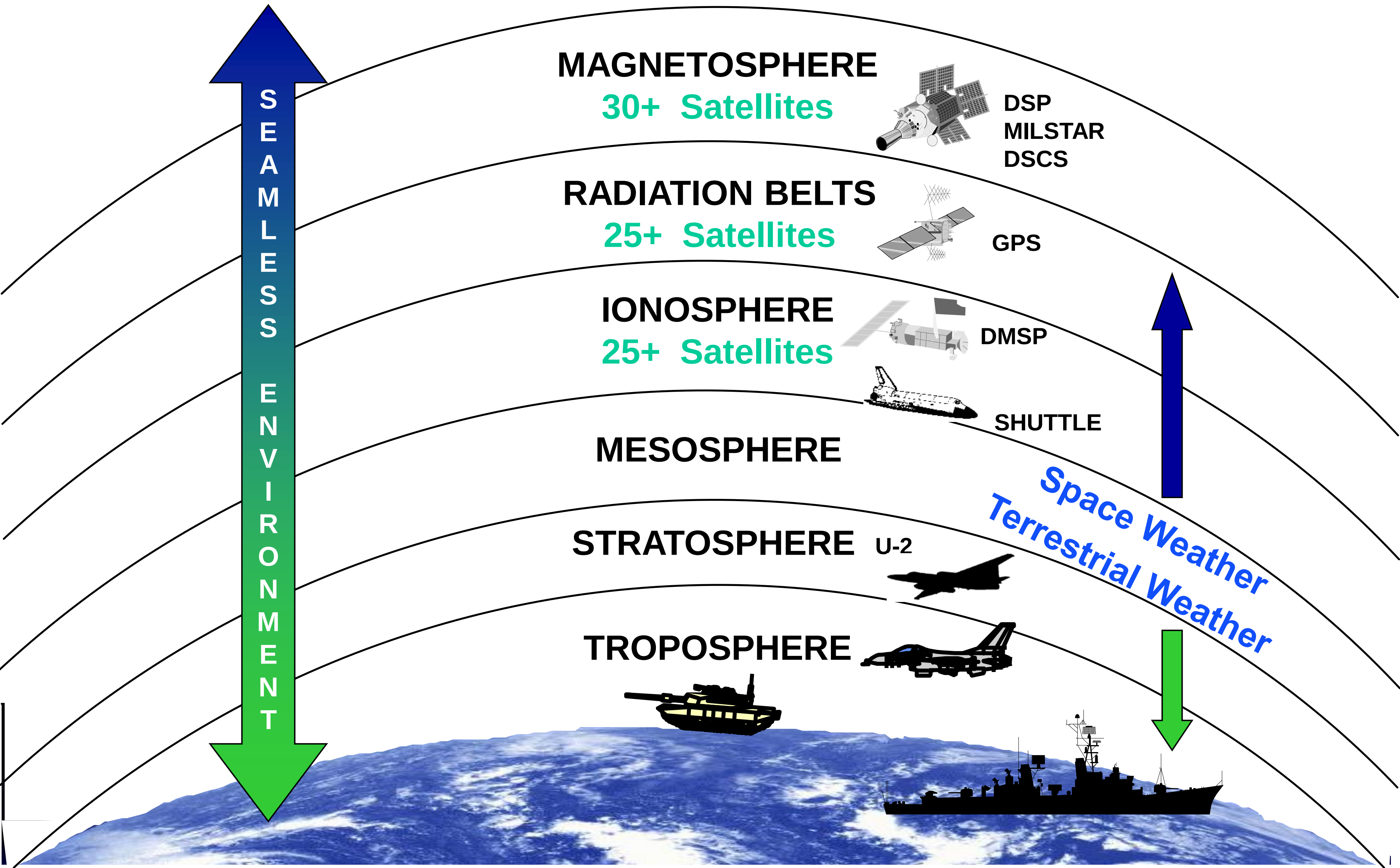
Systems, <u>and what else</u>?

The environment continues with defining the context within which systems *interact*; and that context can advantage, or disadvantage, them ...

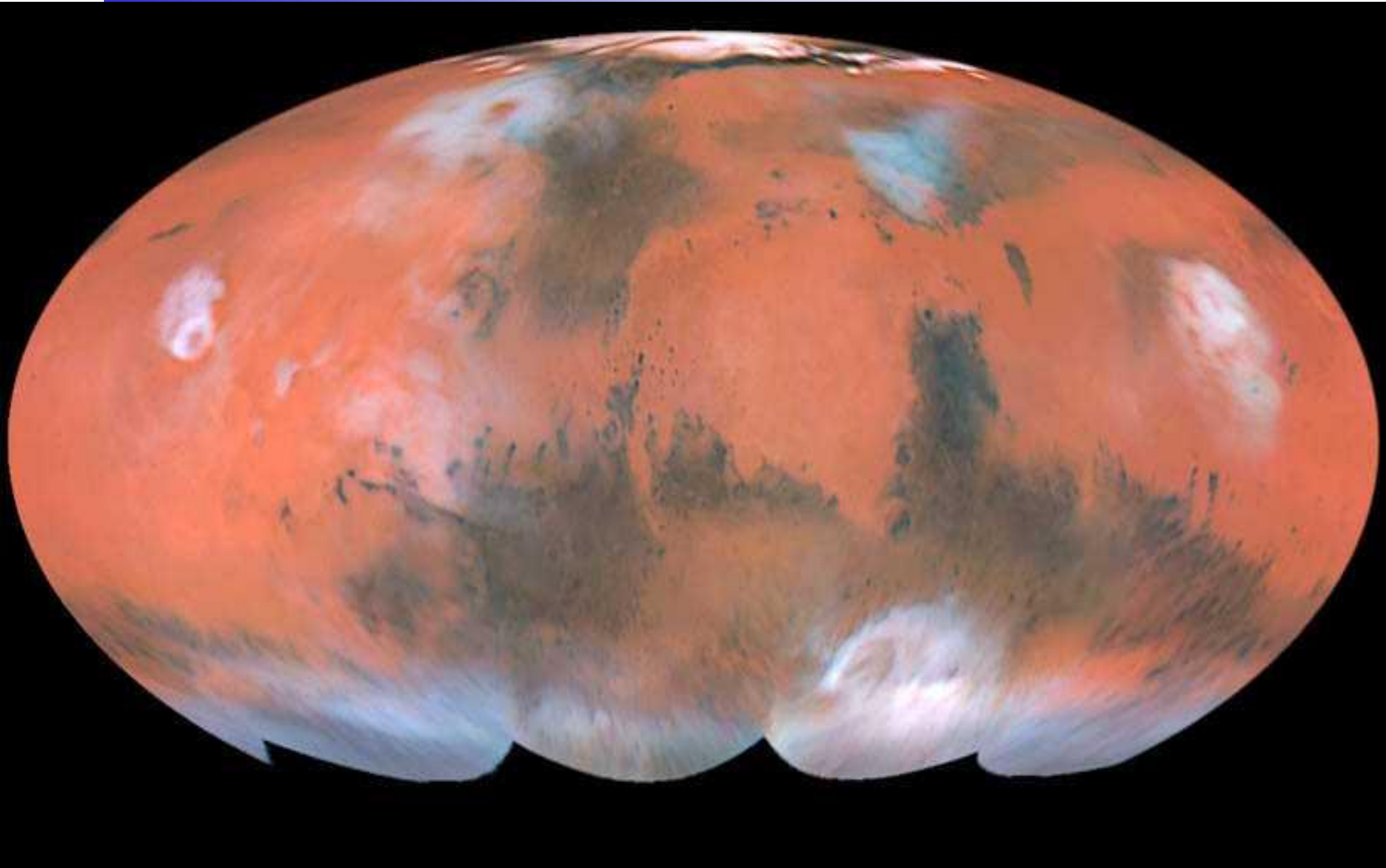
The scope of the SRM

The SRM is developed to include spatial referencing for any real or conceptual object in the solar system, or any abstract or virtual universe.

It's a Loooong way up



The SRM encompasses more than the Earth



The SRM requires a shared solution

- If you have spatially located data,
  - Who doesn't?
- Then you need to understand spatial referencing,
  - A very complex topic (as we'll demonstrate)
- Especially if you intend to interchange data.

- This situation/problem is not unique to M&S.
  - *E.g.*, precise positioning for navigation, range tracking,…

- A "worldwide problem"
  - Being tackled by ISO/IEC JTC 1/SC 24/WG 8.
  - ISO/IEC 18026: Spatial Reference Model (SRM)
  - See the following URLs for more information:

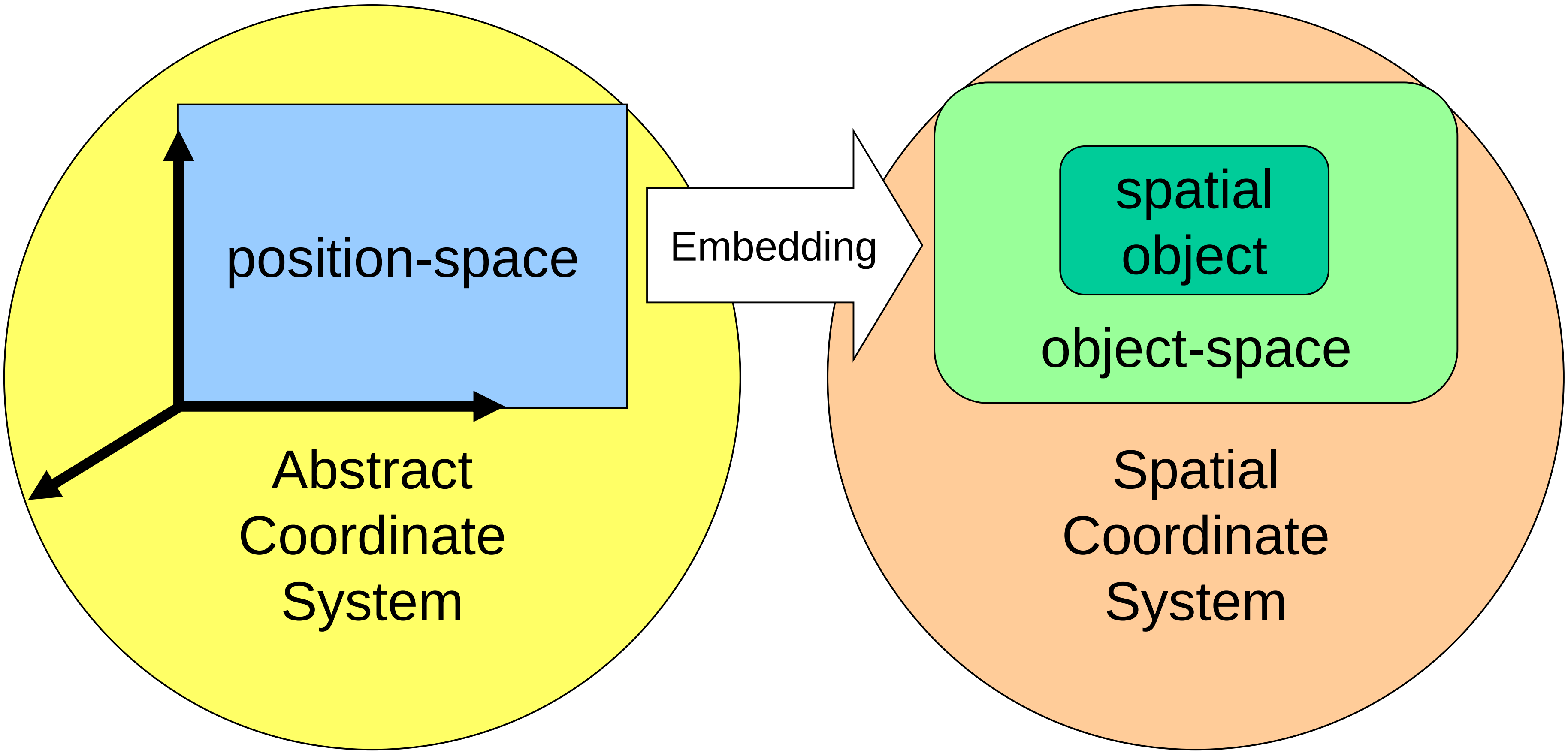
    http://www.iso.ch
    http://www.jtc1.org
    *h*ttp://isotc.iso.ch/livelink/livelink.exe?func=ll&objId=327973&
    objAction=browse&sort=name
    http://www.sedris.org/wg8home

SRM goals

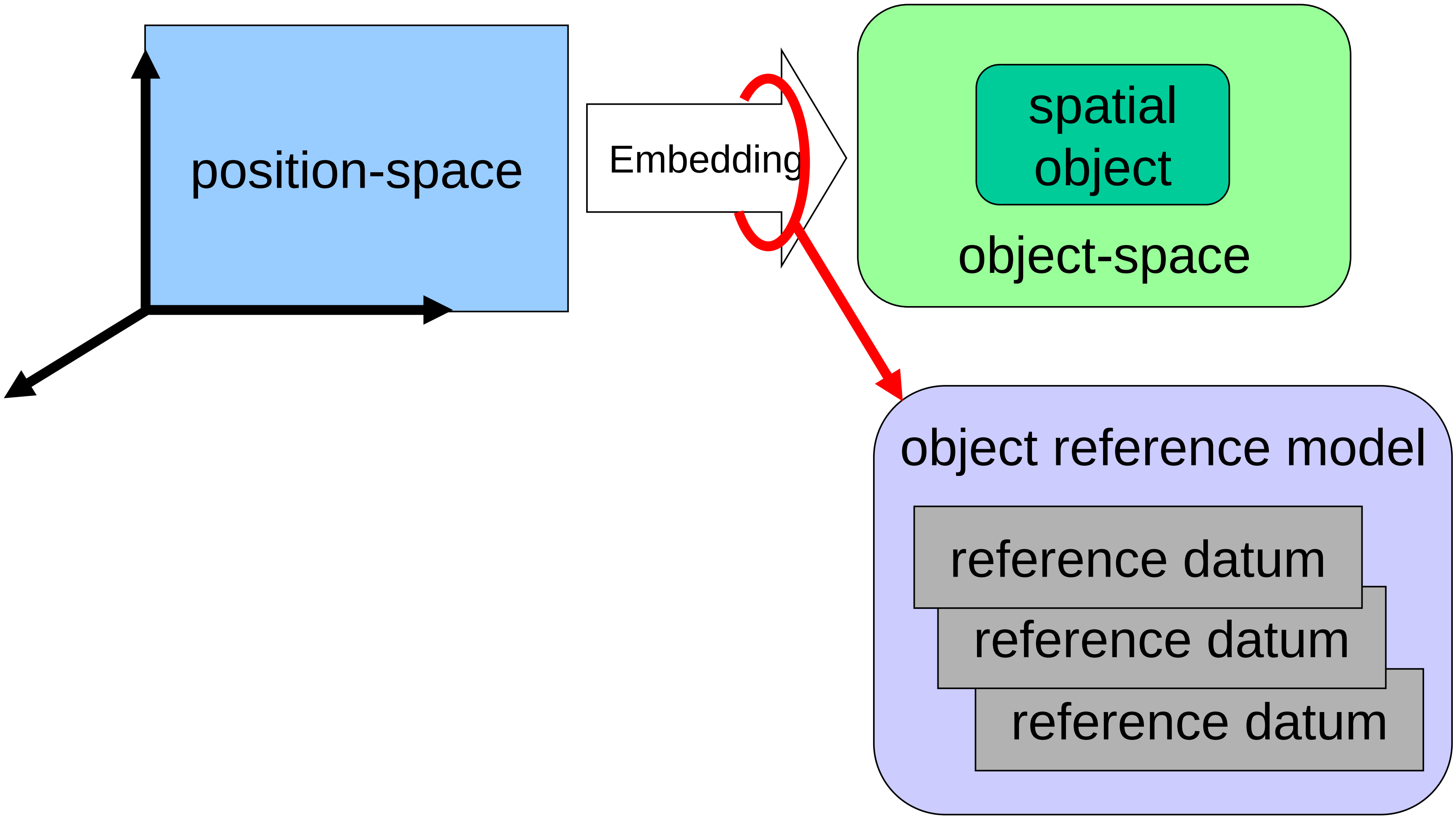
- Unambiguous Spatial Referencing
  - locations and directions
  - well defined terminology
- A single framework for diverse communities of users
  - CAD/CAM
  - modeling and simulation
  - geodesy
  - space science
- Extensibility
  - including an extensible API
  - a standardized registration process

SRM concepts

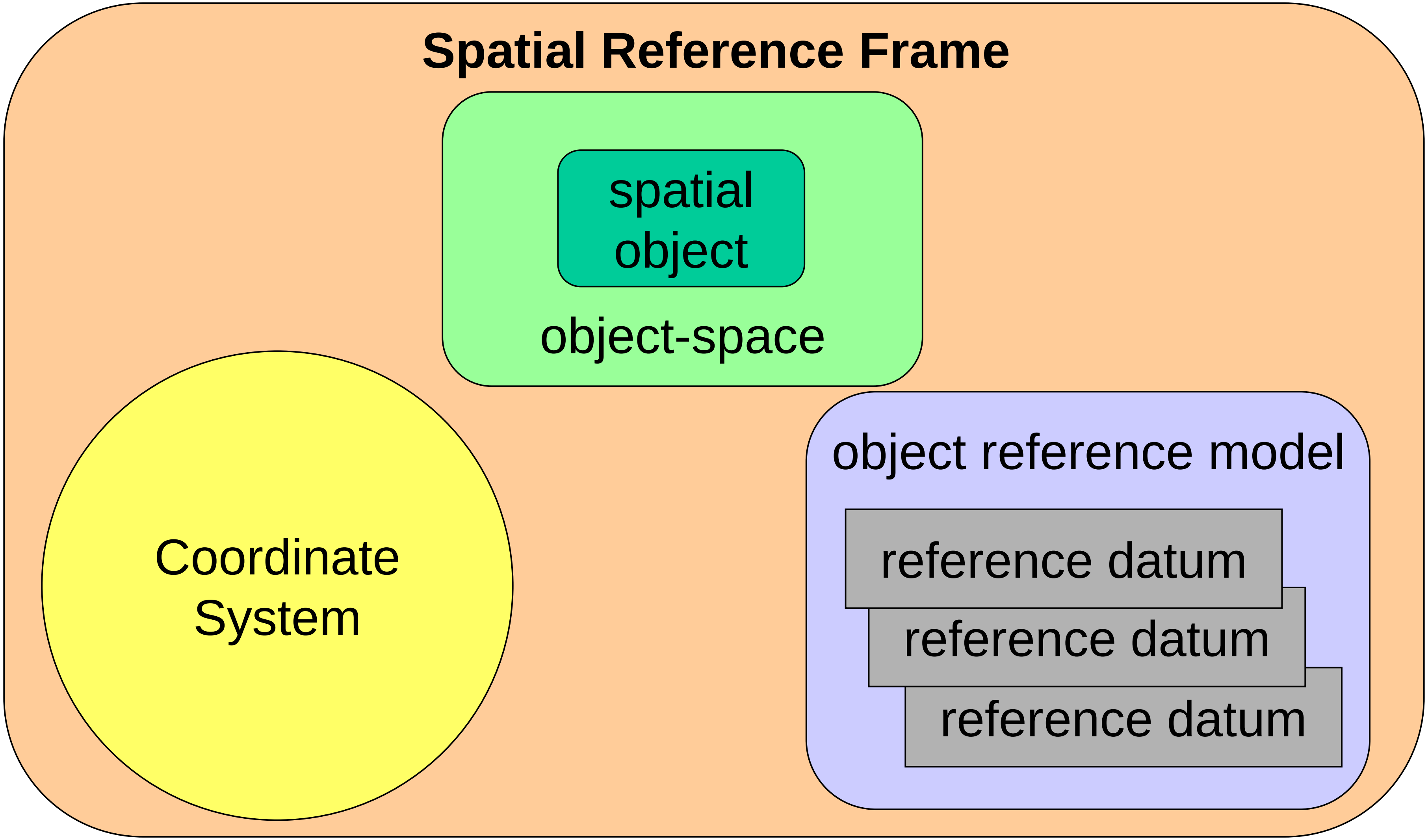
Key concepts
(Preview)



Key concepts
(Preview)



Key concepts
(Preview)



SRM concepts

- **Spatial objects and object-space**
- Position-space and normal embeddings
- Reference datums
- Object reference models
- Coordinate systems
  - Abstract coordinate systems
    - localization
    - Induced surface coordinate systems
    - map projections
  - Temporal coordinate systems
  - Spatial coordinate systems
- Spatial reference frames
- Vertical offset surfaces
- Spatial operations

Spatial objects and object-space

spatial object

Object-space:

The real or abstract universe that contains a spatial object.

(A spatial object is assumed to be fixed in its object-space.)

Spatial objects and object-space

spatial object

- physical object
- abstract object

"real world"

engineering models,
virtual world,
CAD,...

The length of a metre has intrinsic meaning.

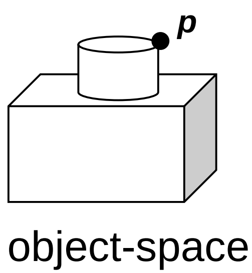
The length of a metre length does **not** have intrinsic meaning.

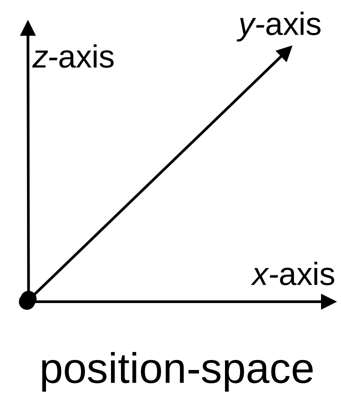
Length scale **shall** be defined.

SRM concepts

- Spatial objects and object-space
- **Position-space and normal embeddings**
- Reference datums
- Object reference models
- Coordinate systems
  - Abstract coordinate systems
    - localization
    - Induced surface coordinate systems
    - map projections
  - Temporal coordinate systems
  - Spatial coordinate systems
- Spatial reference frames
- Vertical offset surfaces
- Spatial operations

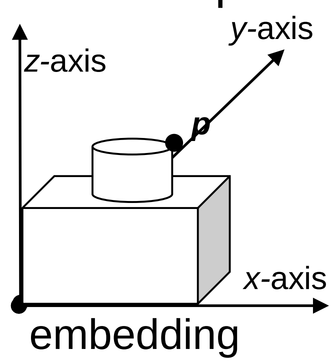
Position-space and embeddings





Position-space:

An *n*-dimensional Euclidean space abstraction of object-space.

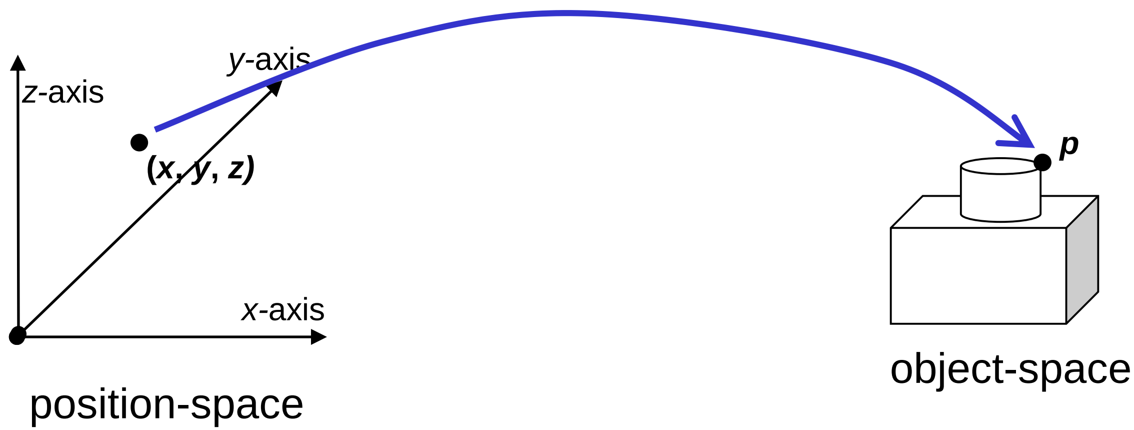


embedding:

A particular position-space model of object space.

Position-space and embeddings

An embedding is a function from position-space to object-space.

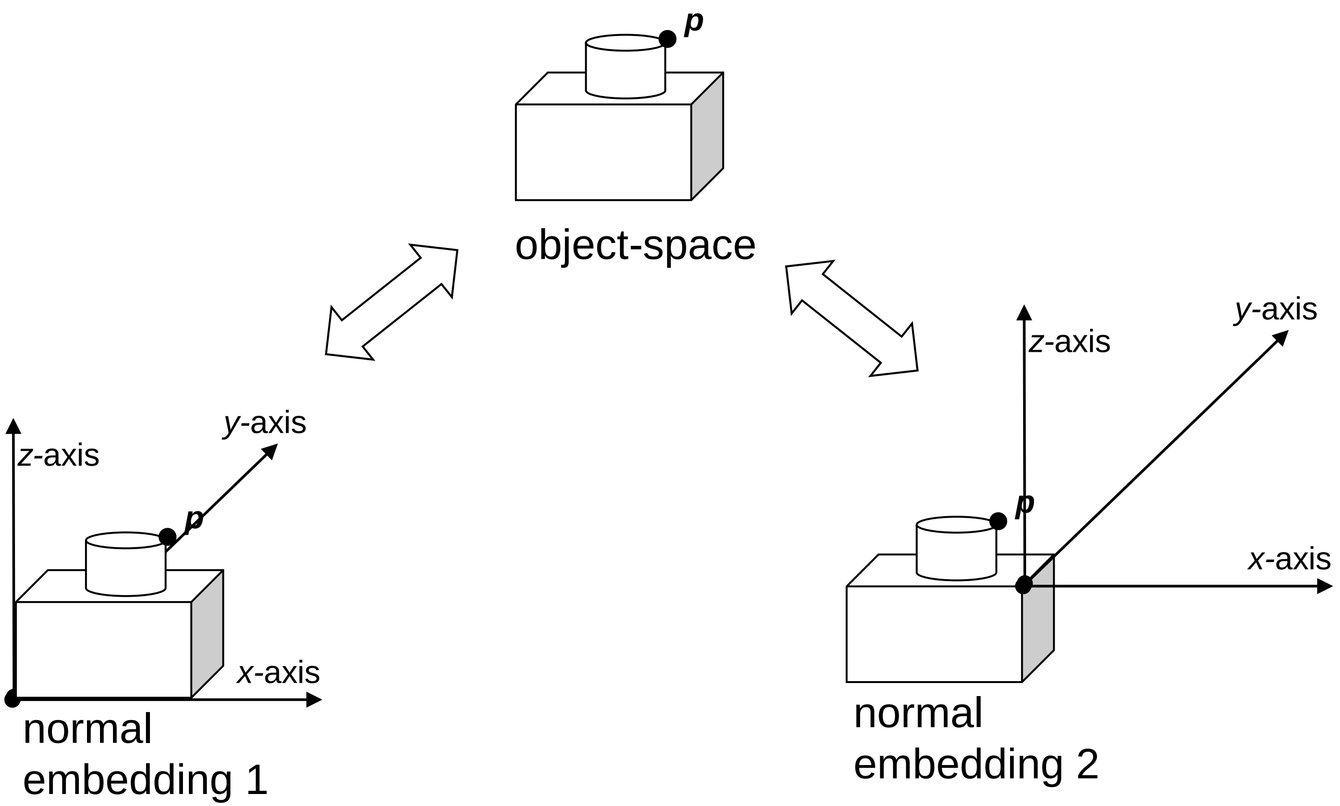


An embedding is **normal** if distance is preserved assuming
1 Euclidean unit = 1 metre in object-space.

In the 3D case, an embedding is either right-handed or left-handed.

An embedding allows position-space operations and methods of algebraic geometry to be applied to object-space.

Position-space and embeddings



There are infinitely many normal embeddings of a given object-space.

In general, no one normal embedding is intrinsic or canonical.

Position-space and embeddings

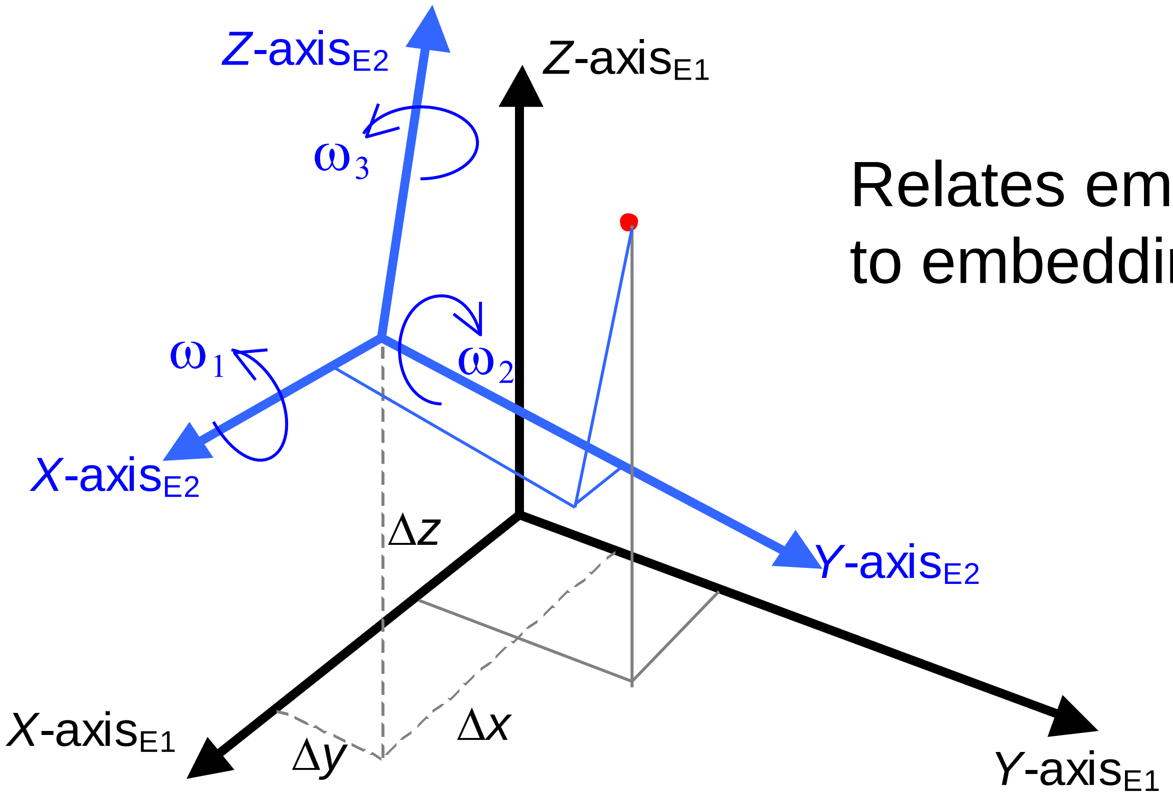
Given two normal embeddings, E_1 and E_2 how are are they related to each other?

7-parameter transformation:

$$\begin{pmatrix} x \\ y \\ z \end{pmatrix}_{E1} = \begin{pmatrix} \Delta x \\ \Delta y \\ \Delta z \end{pmatrix}_{E1} + (1+\Delta s)\,\mathbf{T_3 T_2 T_1} \begin{pmatrix} x \\ y \\ z \end{pmatrix}_{E2}$$

Each $\mathbf{T}_k$ is a rotation matrix ...

7 parameter transformation



Relates embedding E2 to embedding E1

SRM concepts

- Spatial objects and object-space
- Position-space and normal embeddings
- **Reference datums**
- **Object reference models**
- Coordinate systems
  - Abstract coordinate systems
    - localization
    - Induced surface coordinate systems
    - map projections
  - Temporal coordinate systems
  - Spatial coordinate systems
- Spatial reference frames
- Vertical offset surfaces
- Spatial operations

How are embeddings specified?

- The specification pieces
  - Reference Datum (RD)
  - Spatial binding
- RD and normal embedding compatibility
- “Assembly instructions”:
  - Object Reference Model Template (ORMT)
- Object Reference Model (ORM)
  - ORMT Realization
  - Embedding specification

Reference datums

Reference datum (RD):

A geometric construct in position-space that is used to specify an aspect of an embedding of position-space into object-space.

Reference datum (RD)

- Concept
  - A construct with position-space analytic representation and object-space geometric representation
  - Category:
    - Point
    - Directed curve
    - Oriented surface
  - Expressed as implicit or explicit function of position-space
  - Method of orientation (except point)
  - <u>Method</u> of binding to object is application domain specific and is outside of SRM scope
- Standardized set
  - Label, code, definition
  - set of Ellipsoid RD associated to celestial objects

Reference datum (RD)

RD Categories				
Category name	**Position-space representation**			**Spatial representation**
	1D	**2D**	**3D**	
Point	(a) real a	(a, b) real a, b	(a, b, c) real a, b, c	a position in object-space
Directed Curve		Curve specification $\mathbf{p} = \mathbf{F}(t)$, $\mathbf{F}$ is smooth and $\mathbf{R}^2$ valued. Direction at $\mathbf{p_0} = \mathbf{F}(t_0)$ is $\mathbf{n} = \frac{d\mathbf{F}}{dt}(t_0)$.	Curve specification $\mathbf{p} = \mathbf{F}(t)$, $\mathbf{F}$ is smooth and $\mathbf{R}^3$ valued. Direction at $\mathbf{p_0} = \mathbf{F}(t_0)$ is $\mathbf{n} = \frac{d\mathbf{F}}{dt}(t_0)$.	curve in object-space with a designation of direction along the curve
Oriented Surface			Surface generating function $f(\mathbf{p}) = 0$ f is a polynomial of degree 2 or less. Positive side of surface (orientation): $f(\mathbf{p}) > 0$	surface in object-space with a designation of one side as positive.

3D point RDs

RD Label	Description	position-space representation	RD code
RD_3D_ORIGIN	Origin	(0,0,0)	1
RD_3D_X_UNIT_POINT	*X*-axis unit point	(1,0,0)	2
RD_3D_Y_UNIT_POINT	*Y*-axis unit point	(0,1,0)	3
RD_3D_Z_UNIT_POINT	*Z*-axis unit point	(0,0,1)	4

3D directed curve RDs

RD Label	Description	position-space representation	RD code
RD_3D_X_AXIS	*X*-axis	$\mathbf{F}(t) = t(1,0,0)$	5
RD_3D_Y_AXIS	*Y*-axis	$\mathbf{F}(t) = t(0,1,0)$	6
RD_3D_Z_AXIS	*Y*-axis	$\mathbf{F}(t) = t(0,0,1)$	7

Oriented surface RD (planes)

RD label	Description	Position-space representation	RD code
RD_3D_XY_PLANE	*xy*-plane	$0 = f(x, y, z) \equiv z$	8
RD_3D_XZ_PLANE	*xz*-plane	$0 = f(x, y, z) \equiv y$	9
RD_3D_YZ_PLANE	*yz*-plane	$0 = f(x, y, z) \equiv x$	10

Oriented surface RD (ellipsoids)

$$0 = f(x,y,z) = \frac{x^2}{a^2} + \frac{y^2}{a^2} + \frac{z^2}{b^2} - 1.$$

- Oblate: $a > b$
- Sphere: $a = b$
- Prolate: $a < b$

Example:

RD_AIRY Airy 1830

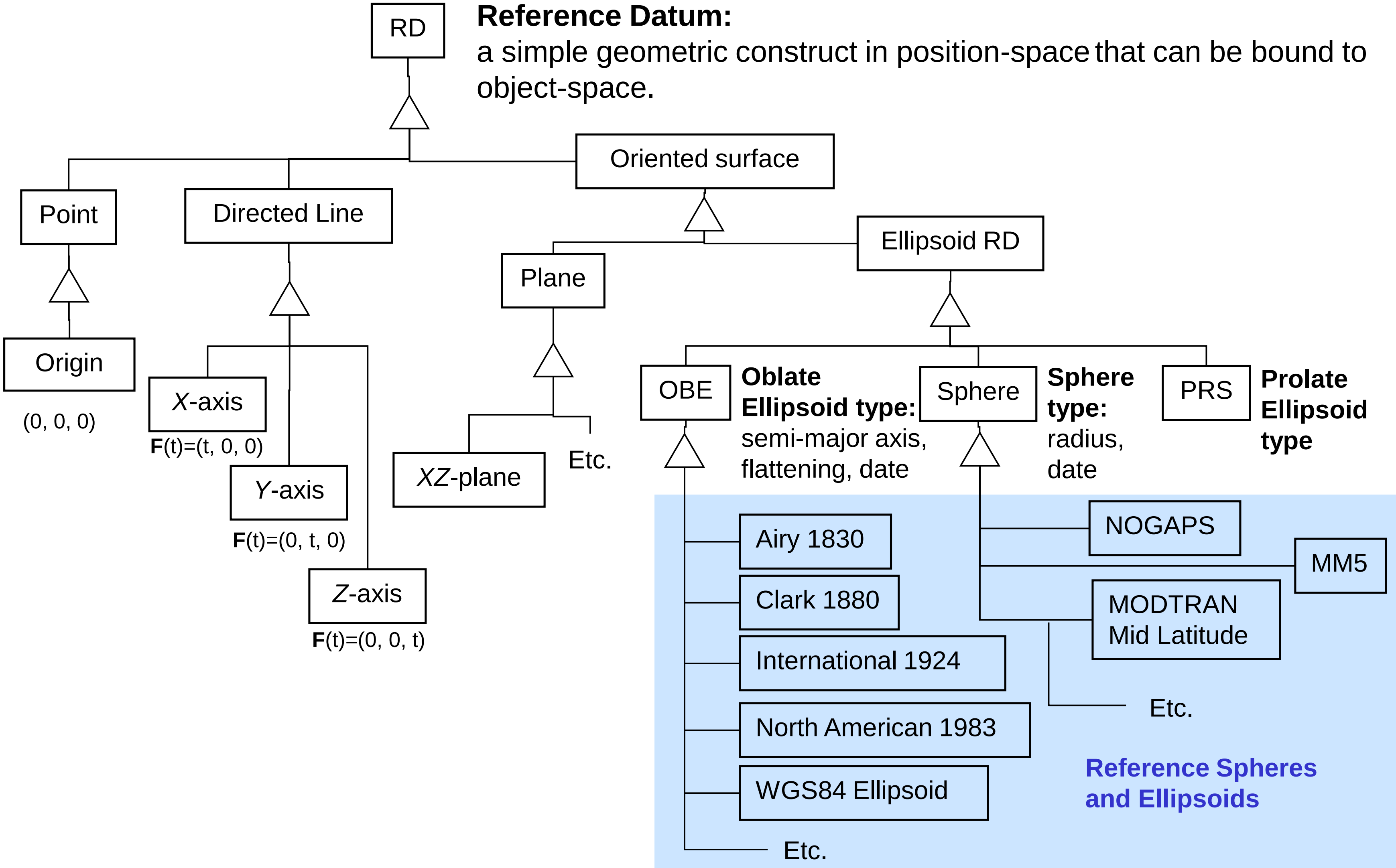
a = 6 377 563,396 flattening = 1/299,324 964 6

flattening = (a-b)/a

b = a·(1-flattening)

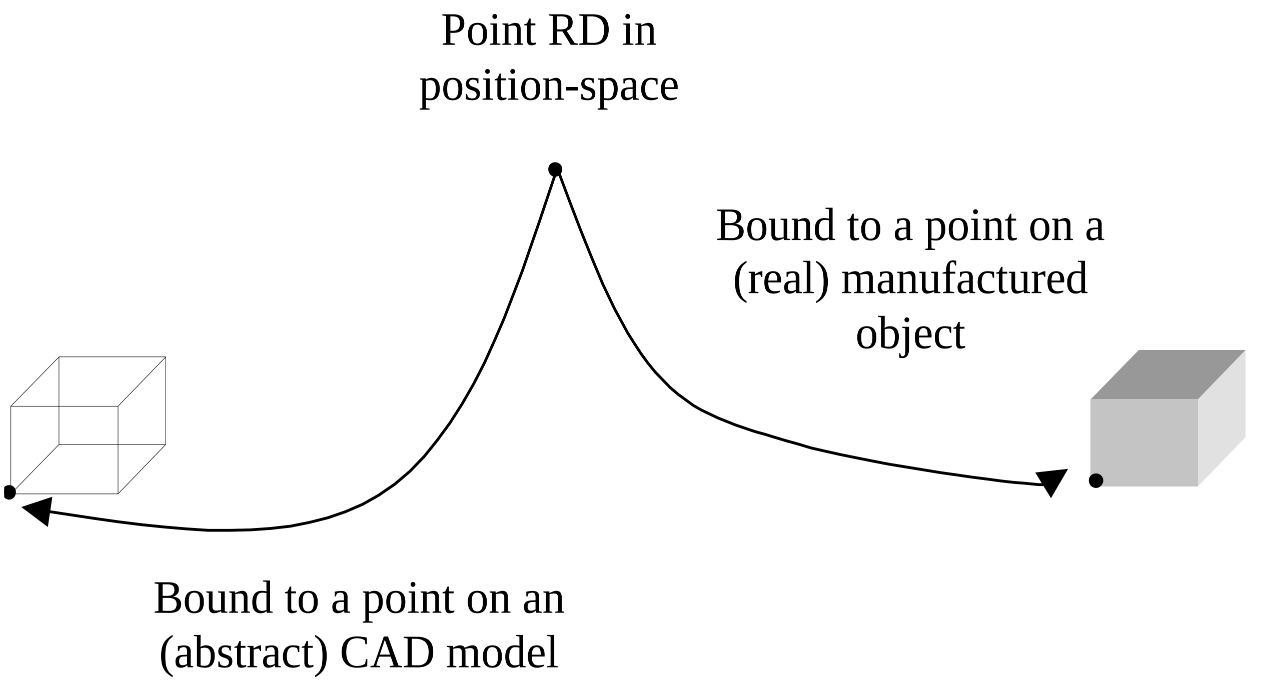
Standard reference datums

Reference Datum:
a simple geometric construct in position-space that can be bound to object-space.



RD bindings

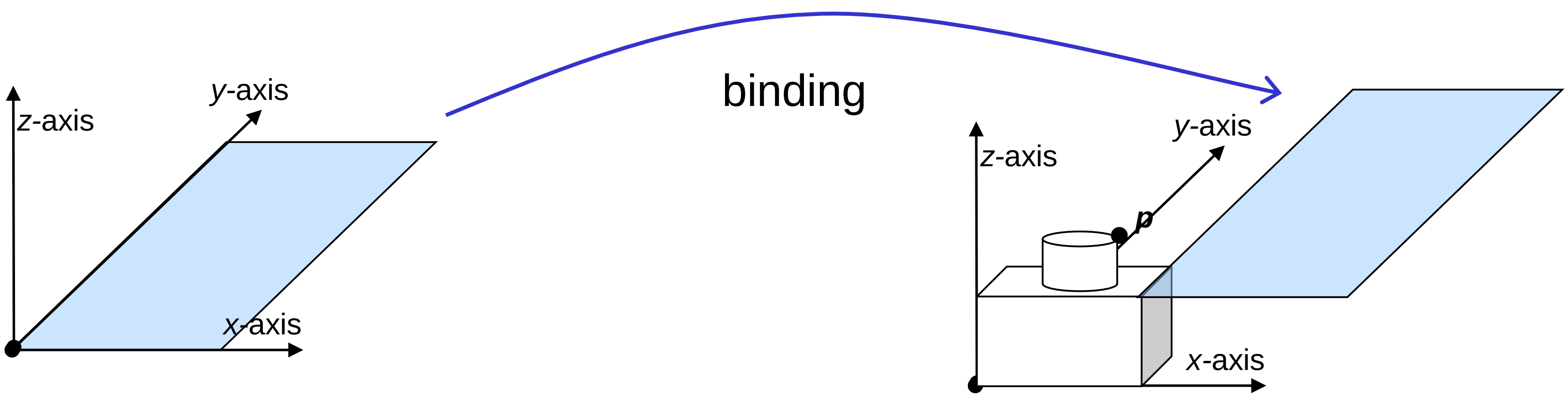
An RD is **bound** when the RD is identified with a corresponding constructed entity in object-space.



Points are bound to identified points, directed curves to constructed curves, and oriented surfaces to constructed surfaces.

RD bindings and embedding compatibility

An RD is **compatible** with a normal embedding if the locus of the RD in position-space is coincident with the points (and direction or orientation, as applicable) of the geometric construction of the binding.

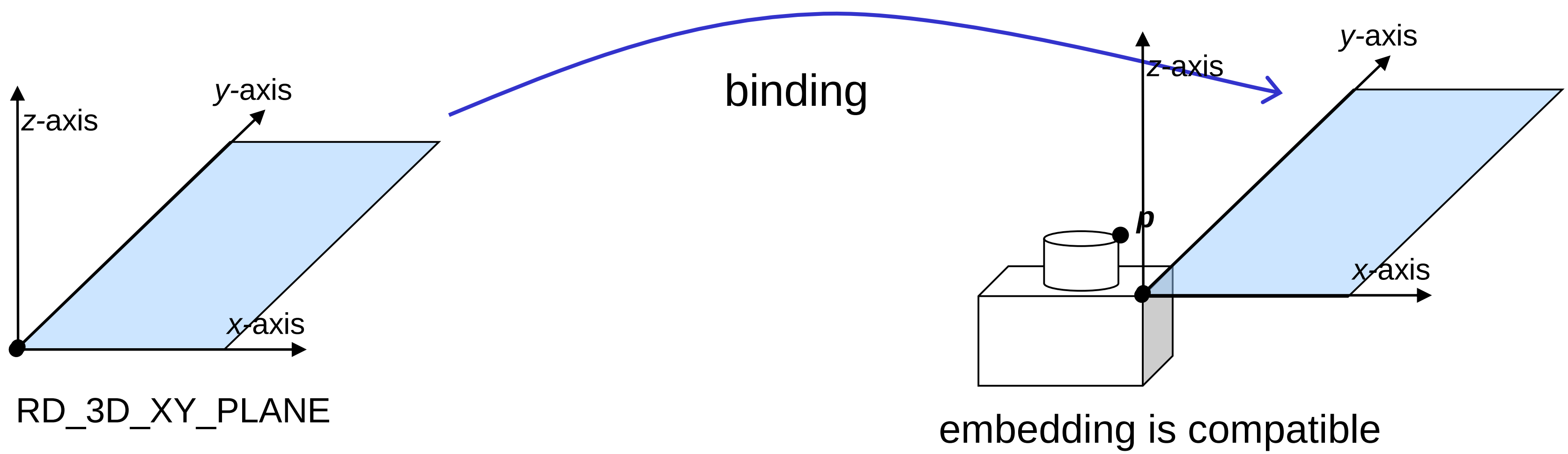


RD_3D_XY_PLANE

embedding is **not** compatible

RD bindings and embedding compatibility

An RD is **compatible** with a normal embedding if the locus of the RD in position-space is coincident with the points (and direction or orientation, as applicable) of the geometric construction of the binding.



Object reference models

- The set of normal embeddings compatible with a *single* RD binding has many members.

- The set of normal embeddings compatible with a set of 2 or more RD bindings may be empty.

- A careful selection of RDs and bindings may result in a compatible set of normal embeddings consisting of a *single* member.
  I.e.: There is a *unique* compatible embedding for the bound RD set.

- This is the concept for an *object reference model.*

Object reference models

Object reference model (ORM) for a spatial object:

A set of RDs bound by identification with geometric constructions in object-space for which there exists exactly one normal embedding of position-space into object-space which is compatible with each RD binding in the set.

In the 3D case, this unique embedding shall also be right-handed.

Object reference model template

- An ***object reference model template (ORMT)***, is a set of RD components, together with binding constraints which will determine a unique normal embedding of position-space into object-space when an instance of each the RD components is bound to corresponding geometric constructs in object-space that conform to the binding constraints.
  - If the position-space is 3D, the unique normal embedding shall be right-handed.

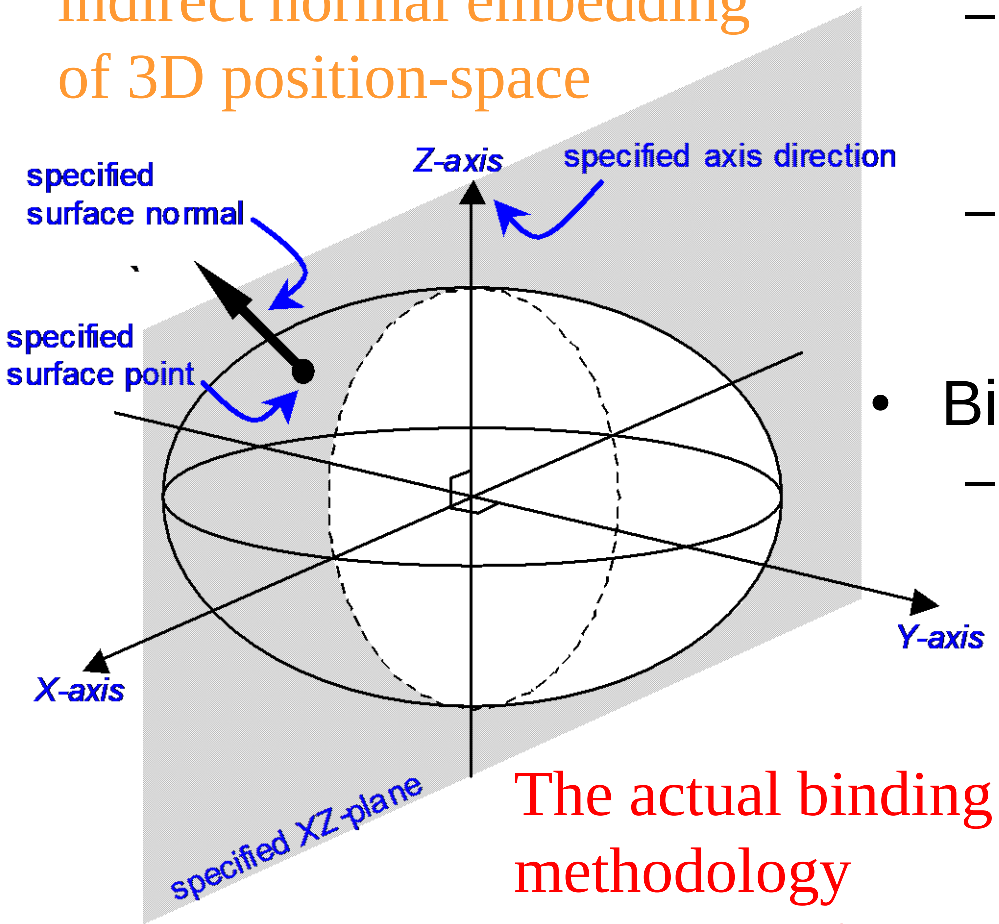
- A ***binding constraint*** is a specification of relationships in object-space between geometric constructs corresponding to RD instances.
  - The binding constraint specification may include containment of a point in a curve, or a surface, or a curve in a surface, coincidence of a line with the axis of symmetry of a surface, the right-handedness of sets of directed lines or oriented planes, or the distance in object-space between points.

ORMT example

ORMT LABEL	ORMT_3D_SPHERE
Description	3D sphere with designated directional axis and xz-plane
RDs	RD 1. The sphere RD with radius *r*. RD 2. RD_3D_Z_AXIS RD 3. RD_3D_XZ_PLANE
Binding constraints	BC 1. The constructed directed line bound to RD 2 shall contain the centre of the constructed sphere bound to RD 1. BC 2. The constructed plane bound to RD 3 shall contain the constructed directed line bound to RD 2. BC 3. The radius of the constructed sphere bound to RD 1 shall be *r* (scaled) metres.

Realization of an ORMT

The binding creates an indirect normal embedding of 3D position-space



The actual binding methodology is **not** part of the SRM

- RDS
  - RDS
    - Oblate ellipsoid instance
    - XZ-plane
  - Binding Rule
    - the ellipsoid minor axis must be contained in the plane.
- Binding
  - There is only one place in Earth's space for the RDS with:
    - ellipsoid minor axis points parallel to Earth's rotational axis
    - The point is on the ellipsoid surface with given long/lat
    - The surface normal at the point points in the given direction.

3D ORMTs

- ORMT_3D_SPHERE
  - 3D sphere with designated directional axis and *xz*-plane
- ORMT_3D_OBLATE_ELLIPSOID
  - 3D oblate ellipsoid with designated minor axis direction and *xz*-plane
- ORMT_3D_PROLATE_ELLIPSOID
  - 3D prolate ellipsoid with designated major axis direction and *xz*-plane
- ORMT_3D_BI_AXIS_ORIGIN
  - *x*- and *y*-axes determined by directed perpendicular lines passing through the origin
- ORMT_3D_TRI_PLANE
  - Origin determined by the intersection of three planes

standard ORMs

- Each standardized ORM is a realization of a standard ORMT
- Concept
  - ORMT realization bound to a specific object at a specific time
    - Intrinsically determines a unique (right-handed) normal embedding
  - Name of object, definition, RD components
  - Object binding
- Standard/registered ORMs
    - Label, code, definition, region, epoch, etc.
    - Standard ellipsoid RD (if so based)
    - Reference transformation(s) specified with respect to the object reference ORM by 7 parameter values:
      - » origin offset (Δx, Δy, Δz)
      - » axis rotations (ω_1, ω_2, ω_3)
      - » scale adjustment Δs
- Every spatial object has one reference ORM
    - The WGS 84 ORM is the Earth reference ORM

ORMT realization example

- North American 1927 (NAD 27)
  - ORMT_3D_OBLATE_ELLIPSOID
    - Oblate ellipsoid RD instance: Clark 1866 ellipsoid
      - semi-major axis 6378206.4 m
      - inverse flattening 294.9786982
  - Binding (not part of the SRM specification)
    - Meades Ranch, Kansas on the oblate ellipsoid surface with long=98°32’30.506” W, lat=39°13’26.686” N and surface normal χ= -1.32 ξ=1.93.
  - ORM reference transformation:
    - specification of the binding by 7 parameters w.r.t WGS 84 ORM
      - origin offset: $\Delta x = -8$, $\Delta y = 160$, $\Delta z = 176$
      - axis rotations $\omega_1 = 0$, $\omega_2 = 0$, $\omega_3 = 0$
      - scale adjustment $\Delta s = 0$

Binding categories

A **binding restriction** for an ORMT is an object specific restriction for the binding of a single RD in the RD set of the ORMT.

A **binding category** is an ORMT with:
- a binding category name,
- a set of binding restrictions for the ORMT, and
- object restrictions which delineate the objects for which the binding restrictions apply.

Example: The equatorial inertial dynamic binding category

Name	equatorial inertial
Object restrictions	A planet in the solar system for which its ecliptic plane is distinct from its equatorial plane.
ORMT	ORMT_3D_BI_AXIS_ORIGIN.
Binding restrictions	1) The RD_3D_ORIGIN is the mass-centre of the planet. 2) The RD_3D_X_AXIS points in the direction to the Sun when the planet is at its vernal equinox. 3) The RD_3D_Z_AXIS is parallel to the rotational axis and points north.

ORMs *vs.* object models

Reference Surface:

An ORM is often selected to contain one or more RDs of category oriented surface to correspond to (physical or conceptual) surface(s) which is significant to the modelled spatial object.

Example: an oblate ellipsoid as a ***model*** of the shape of the Earth.

An RD is chosen and its position with respect to the object is bound so that the RD instance is a “best fit” to the object in some application-specific sense.

In particular, if the RD surface is “fitted” to a specific part of the object surface, the ORM is a ***local model***.

If the RD is selected to best fit the entire surface, the ORM is a ***global model***.

Other Surface models:

- **geoid:** gravity equipotential surface that approximates the mean sea level of the Earth.
- **pressure equipotential surface:** equipotential surface of a pressure field.
- **topographic surface:** interface between the solid and liquid/gas portions of a celestial object.

Some of these surfaces are treated in the SRM as ***vertical offset surfaces*** (a later topic), while others are outside of the scope of the SRM.

SRM concepts

- Spatial objects and object-space
- Position-space and normal embeddings
- Reference datums
- Object reference models
- **Coordinate systems**
  - Abstract coordinate systems
    - localization
    - Induced surface coordinate systems
    - map projections
  - Temporal coordinate systems
  - Spatial coordinate systems
- Spatial reference frames
- Vertical offset surfaces
- Spatial operations

Coordinate Systems (CS)

An **abstract CS** is a means of identifying a set positions in m-dimensional position-space ($1 \leq m \leq 3$) that is comprised of:

a. a CS domain,

b. a generating function, and

c. a CS range,

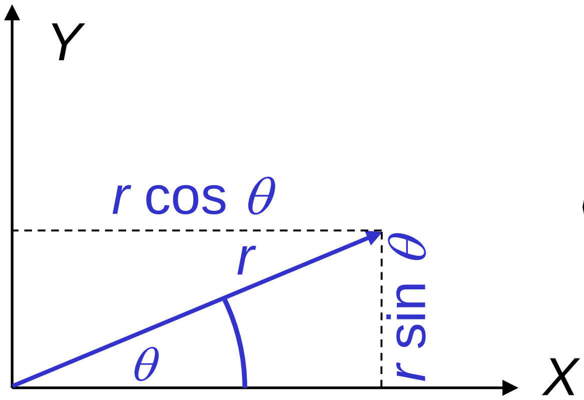
where:

d. the CS domain is a connected domain in the Euclidean space of n-tuples ($1 \leq n \leq m$), called the *coordinate-space,*

e. the *generating function* shall be a one-to-one, smooth, orientation preserving function from the CS domain onto the CS range, and

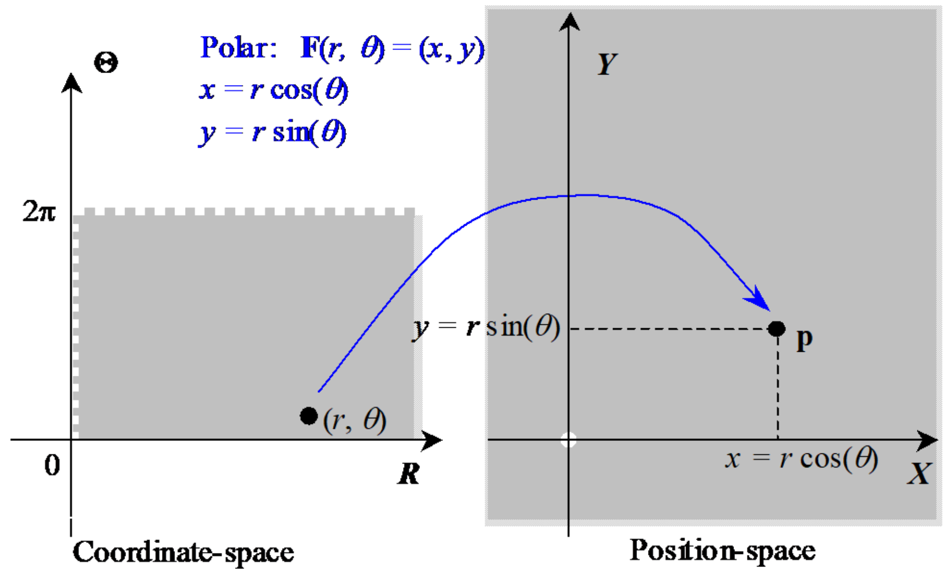
f. the *CS range* is a sub-set of position-space.

Coordinates for position-space



Geometric/Trigonometric derivation

Generating function realization



Not all CSs have Geometric/Trigonometric derivations

Coordinate system type

CS type	coordinate-space dimension	position-space dimension
3D	3	3
surface	2	3
curve	1	3
2D	2	2
plane curve	1	2
1D	1	1

Coordinate system characteristics

A CS is ***linear*** if it has an affine generating function
(a function is ***affine*** if it is a translation operator plus a linear operator)

A non-linear CS is ***curvilinear***.

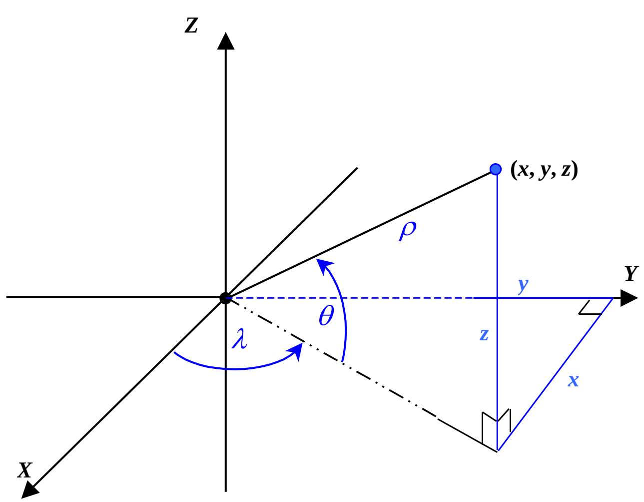
A linear CS is ***orthonormal*** if
<u>3D CS case</u>: the coordinates (1,0,0), (0,1,0), and (0,0,1) are length one and mutually perpendicular in position-space.
<u>Surface and 2D CS case</u>: the coordinates (1,0),and (0,0,1) are length one and mutually perpendicular in position-space.

Spherical CS specification

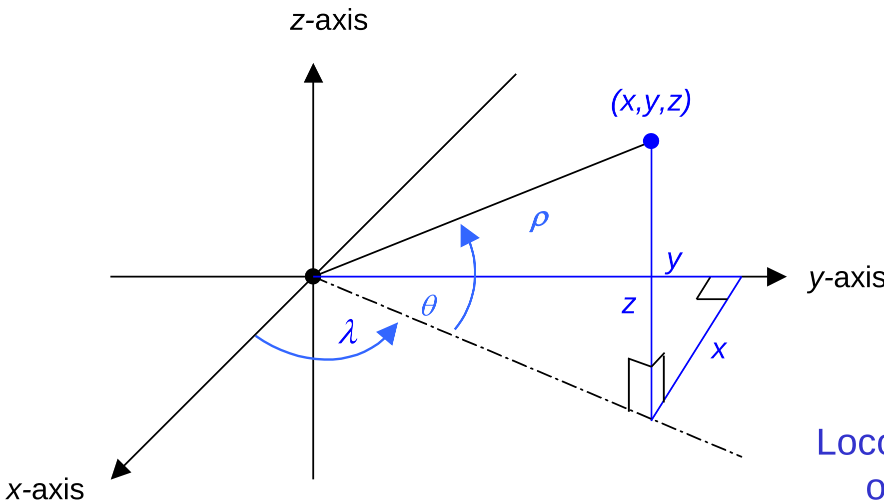
Part 1

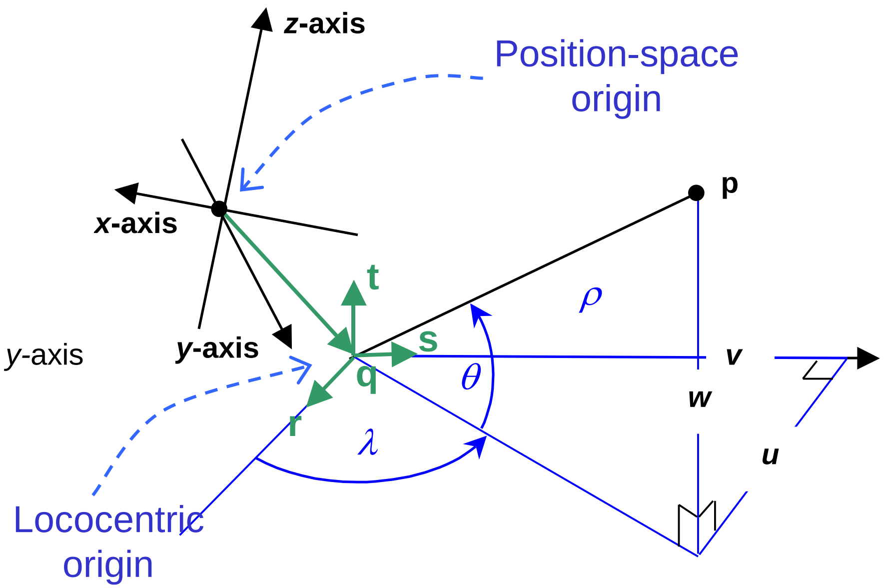
Field	Specification
Common name	Spherical.
Label	CS_SPHERICAL
Function type	Generating function.
CS type	3D.
Properties	Curvilinear, orthogonal.
CS parameters and constraints	None.
Coordinate components	λ : longitude in radians, θ : spherical latitude in radians, ρ : radius
Domain of the generating function or mapping equations	$-\pi/2 < \theta < \pi/2$ $-\pi \leq \lambda < \pi$ $0 < \rho$
Generating function or mapping equations	$\mathbf{F}(\lambda, \theta, h) = (x, y, z)$, where $x = \rho\cos(\theta)\cos(\lambda)$ $y = \rho\cos(\theta)\sin(\lambda)$ $z = \rho\sin(\theta)$

Spherical CS specification
Part 2

Domain of the inverse of the generating function or mapping equations	$\{(x, y, z) \text{ in } \mathbf{R}^3 \mid 0 < x^2 + y^2 + z^2\}$
Inverse of the generating function or mapping equations	$\mathbf{F}^{-1}(x, y, z) = (\lambda, \theta, \rho)$ where $\lambda' = \arctan(y/x)$ principle value, $\lambda = \begin{cases} \lambda' & \text{quadrants I and IV} \\ \pi + \lambda' & \text{quadrants II and III} \end{cases}$ $\rho = \sqrt{x^2 + y^2 + z^2}$, $\theta = \arcsin(z/\rho)$ principle value
Figure(s)	

Lococentric spherical CS





Spherical CS

$$\mathbf{F_s}(\lambda,\ \theta,\ \rho)$$

Lococentric Spherical CS

$$\mathbf{F_{LS}}(\lambda,\ \theta,\ \rho)=\mathbf{L_{3D}}(\mathbf{F_s}(\lambda,\ \theta,\ \rho))$$

Localization operator: $\mathbf{L_{3D}}\left(x,y,z\right)=\mathbf{q}+x\mathbf{r}+y\mathbf{s}+z\mathbf{t},$

where $\mathbf{t}=\mathbf{r}\times\mathbf{s}.$

Coordinate surfaces and curves

3D CS generating function: $\mathbf{G}(u, v, w)$.

Surface 1: $\mathbf{G}_{\mathbf{S1}}(u, v) = \mathbf{G}(u, v, w_0)$.

Surface 2: $\mathbf{G}_{\mathbf{S2}}(u, v) = \mathbf{G}(u, v_0, w)$.

Surface 3: $\mathbf{G}_{\mathbf{S3}}(u, v) = \mathbf{G}(u_0, v, w)$.

Each $\mathbf{G}_{\mathbf{S}k}$ is the generating function for a surface CS.

Curve: $\mathbf{G}_{\mathbf{C1}}(u) = \mathbf{G}(u, v_0, w_0)$.

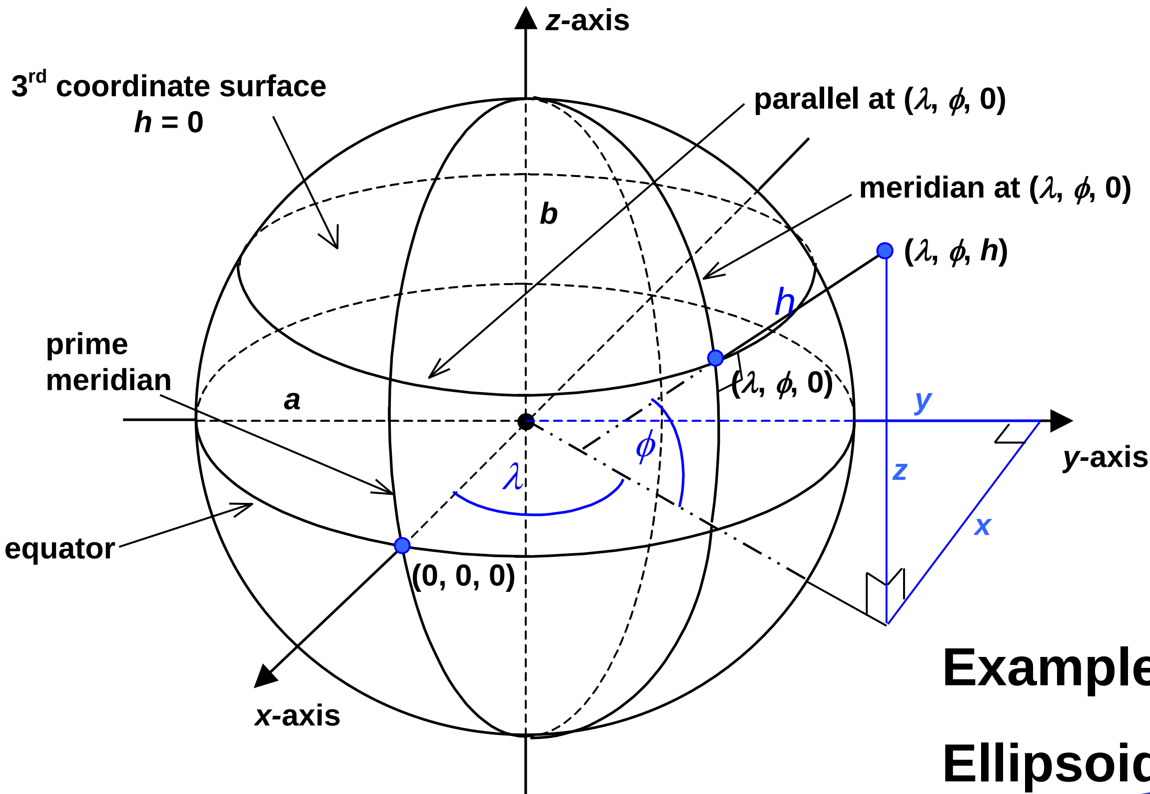
Curve: $\mathbf{G}_{\mathbf{C2}}(v) = \mathbf{G}(u_0, v, w_0)$.

Curve: $\mathbf{G}_{\mathbf{C3}}(w) = \mathbf{G}(u_0, v_0, w)$.

Each $\mathbf{G}_{\mathbf{C}k}$ is the generating function for a curve CS.

A CS is ***orthogonal*** if its coordinate curves intersect at right angles.

Coordinate surfaces and curves



3D geodetic

Example: Geodetic G(λ, ϕ, h)

Ellipsoid: $\mathbf{G_S}(\lambda, \phi) = \mathbf{G}(\lambda, \phi, 0)$.

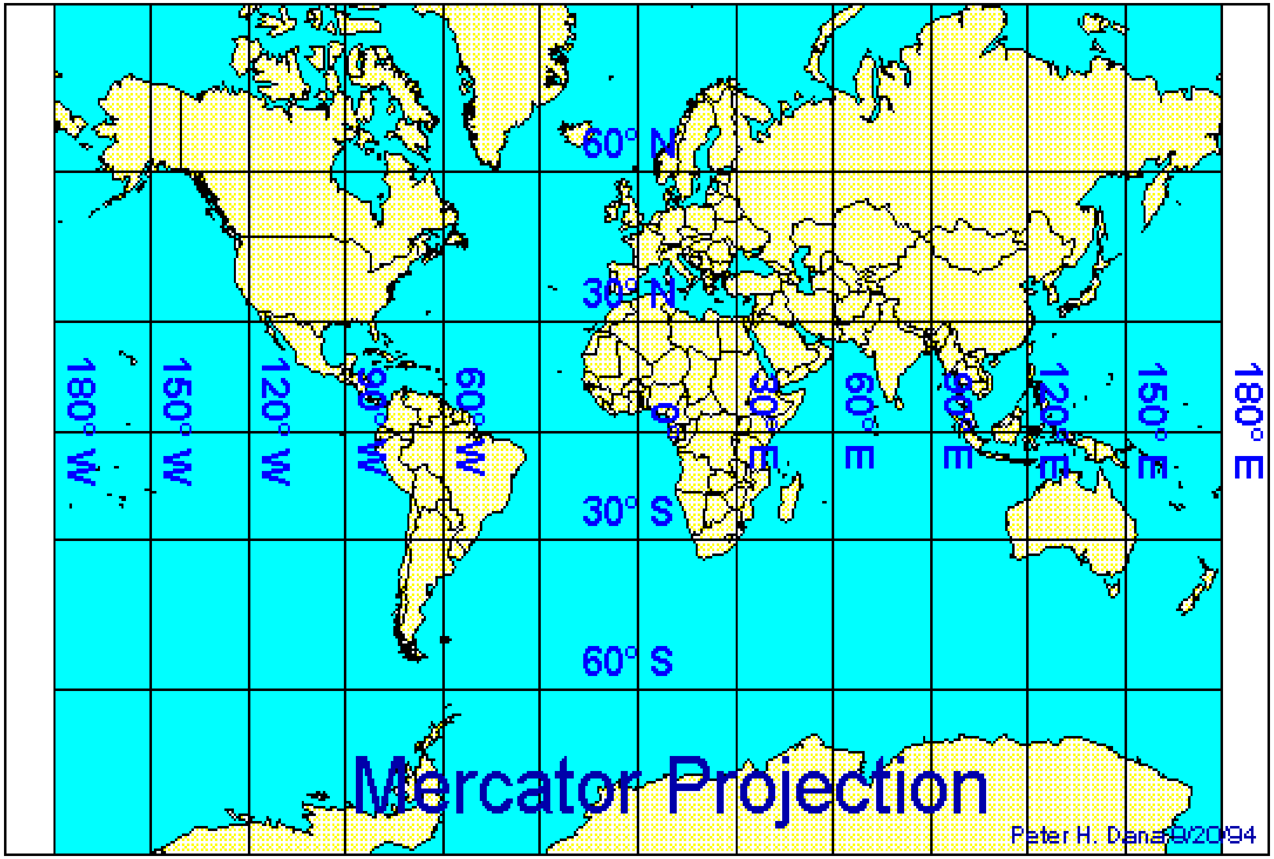
parallel: $\mathbf{G_{C1}}(\lambda) = \mathbf{G}(\lambda, \phi_0, 0)$.

meridian: $\mathbf{G_{C2}}(\phi) = \mathbf{G}(\lambda_0, \phi, 0)$.

(induced)
Surface geodetic

Map projections

Map projections are 2D models of a 3D curved surface.



Map projections

Map projections are 2D models of a 3D curved surface.

A *map projection (MP)* is comprised of

a. a connected region of the surface of an oblate ellipsoid,
b. a generating projection, and
c. an MP range in 2D coordinate-space,

where:

d. the MP range is connected, and
e. the generating projection is one-to-one from the region of the oblate ellipsoid onto its MP range and its inverse function is smooth and orientation preserving.

The generating projection is formulated in terms of surface geodetic coordinates called ***mapping equations*****.**

Map projections

Map projections are 2D models of a 3D curved surface.

Map projection coordinate-space geometry models/approximates oblate ellipsoid surface geometry.

But the model geometry is distorted.

Map projection distortion

- Length distortion
  - The ratio of map distance to geodesic distance is not constant.
  - “Equidistant” map projections have a constant distance ratio in one direction (E.g.: North/South distances)
- Angular distortion
  - A map projection may or may not preserve angles.
  - A map projection is ***conformal*** if it preserves angles.
- Area distortion
  - The ratio of map area to ellipsoid area may not be constant.
  - “Equi-area” map projections have a constant area ratio.

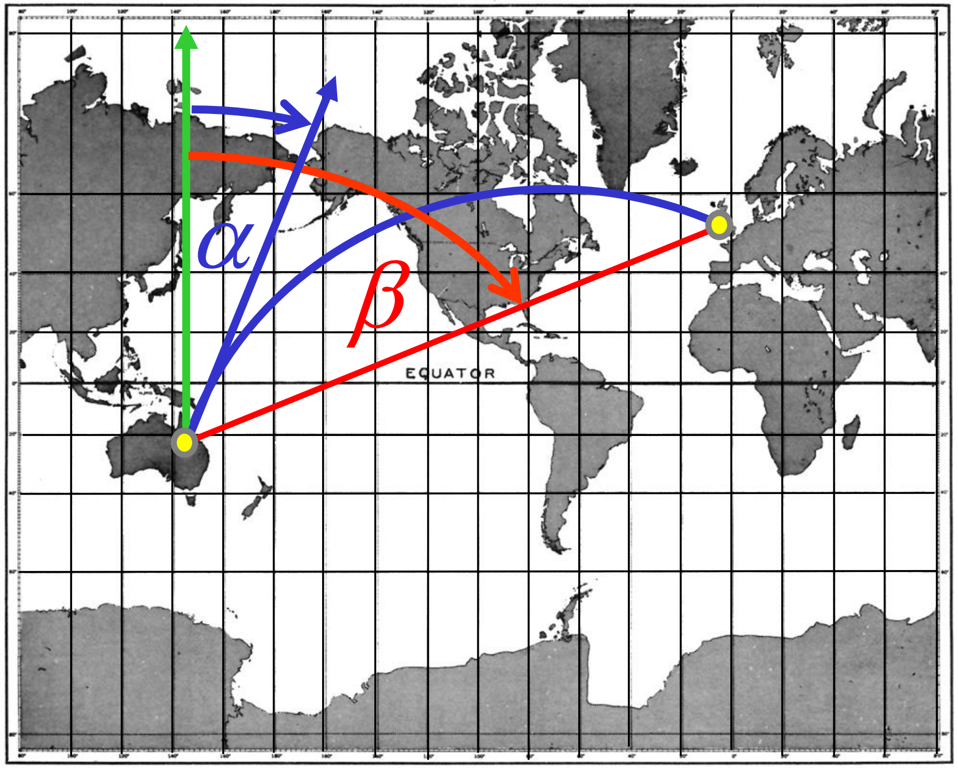
Map projections

Quantifying distortion:

- Map distance *vs* geodesic distance
  - point scale
    - ratio at a point for “infinitesimal” distances
    - varies at different points
    - varies by direction in non-conformal map projections
  - map scale
    - Nominal ratio for the map area
- map azimuth *vs.* geodetic azimuth
- Convergence of the Meridian (COM)

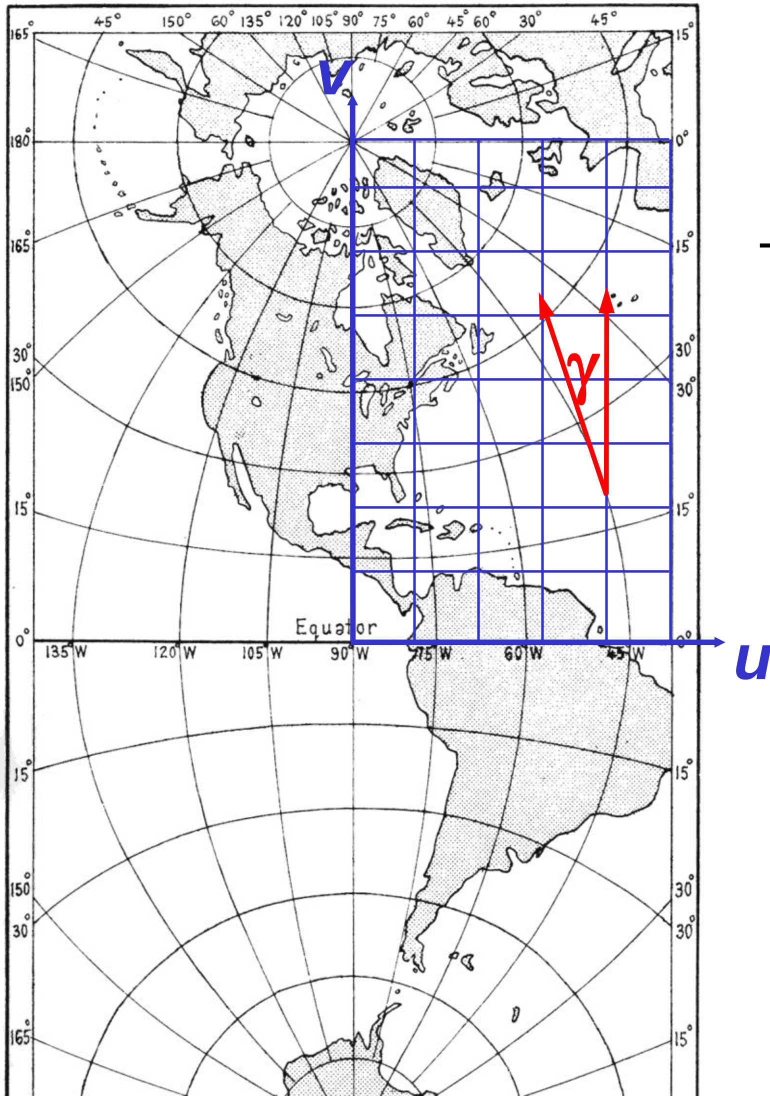
Azimuth

map azimuth β geodetic azimuth α



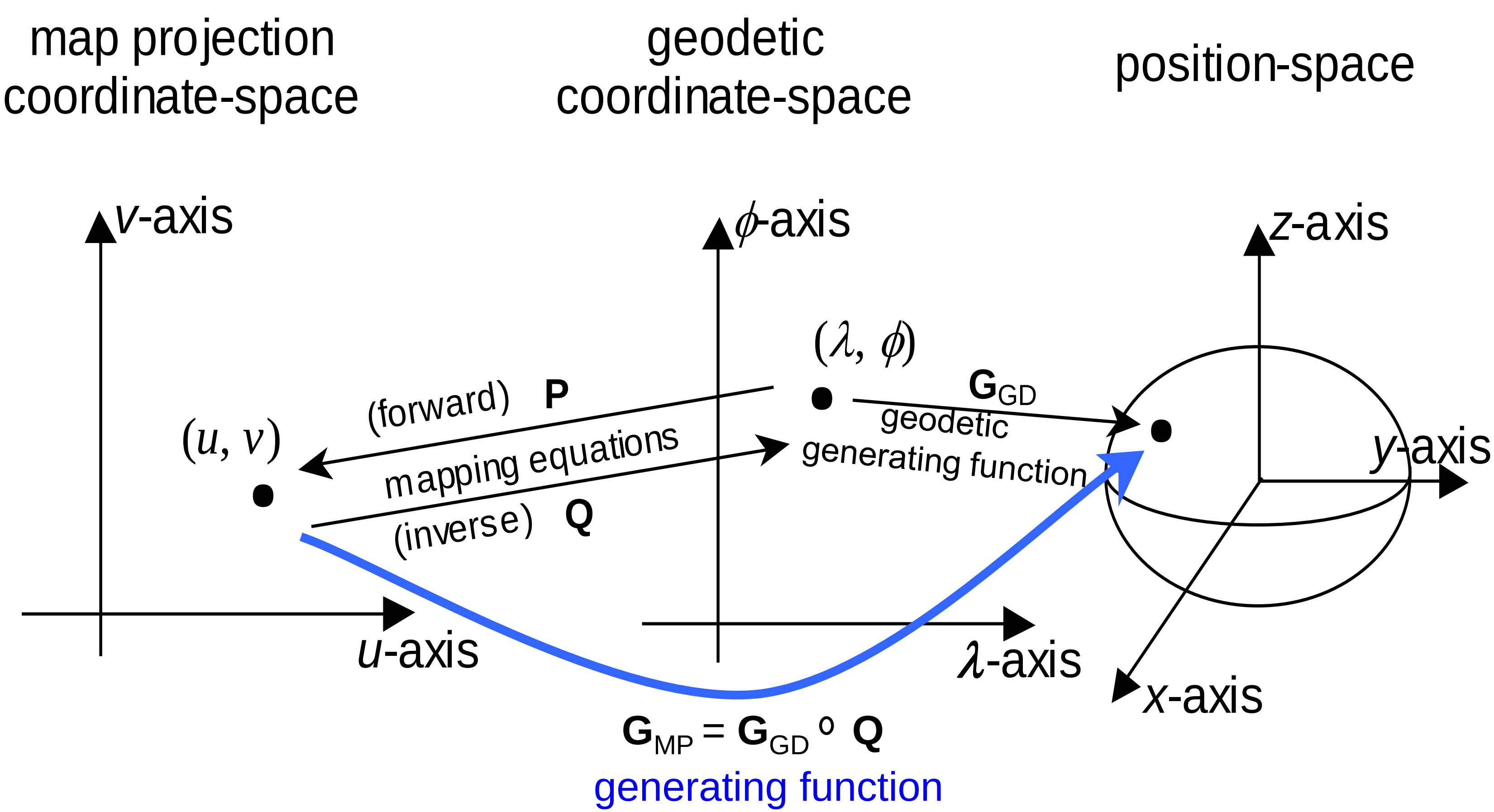
Mercator Map Projection

Convergence of the Meridian



The convergence of the meridian γ is the angle between MP "North" and true North on the ellipsoid (positive clockwise)

Map projection as a surface CS



Augmented map projections

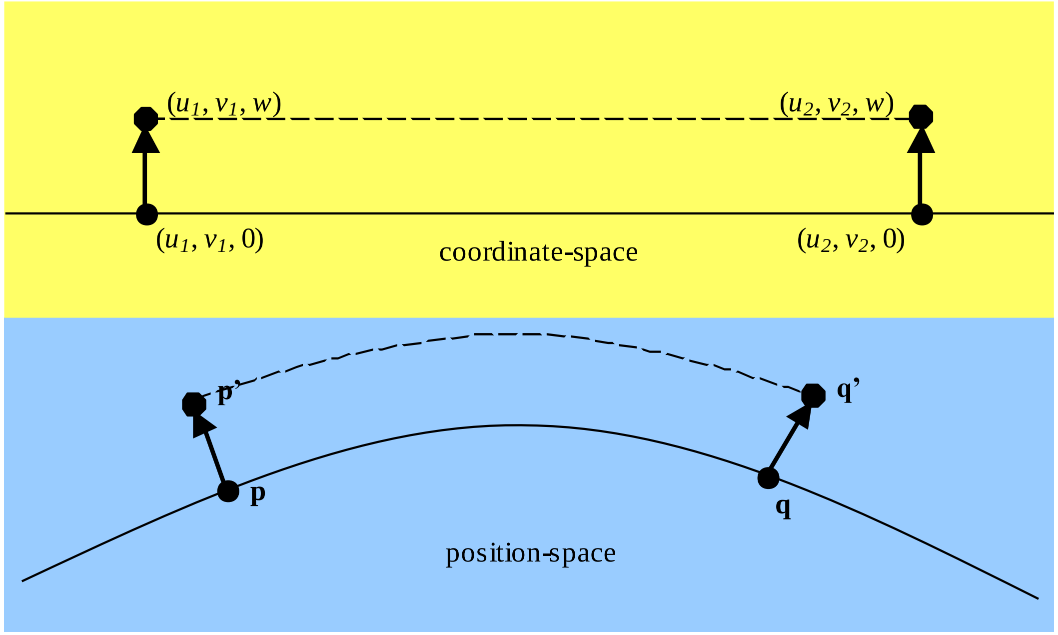
- Simulations usually require three dimensions.
- Some coordinate systems three dimensional by definition.
- Map projections (two surface dimensions) are commonly augmented with a vertical axis to create a three dimensional system.
- Various vertical measures are used for these augmentations, such as:
  - Mean sea level height, orthometric height, geodetic height, pressure altitude, and others.
- These augmentations may be in different units than the other two axes so that the resulting CS may be *vertically distorted*.
- This practice adds additional geometric distortions (now in the third dimension).
- Augmented map projection coordinate-space has a (linear) vector space structure, but the CS is a *curvilinear* CS.

Augmented map projections

Augmenting map coordinates
with ellipsoidal height produces a 3D CS.

MP coordinate (u, v) ⟷ (λ, ϕ) surface geodetic coordinate

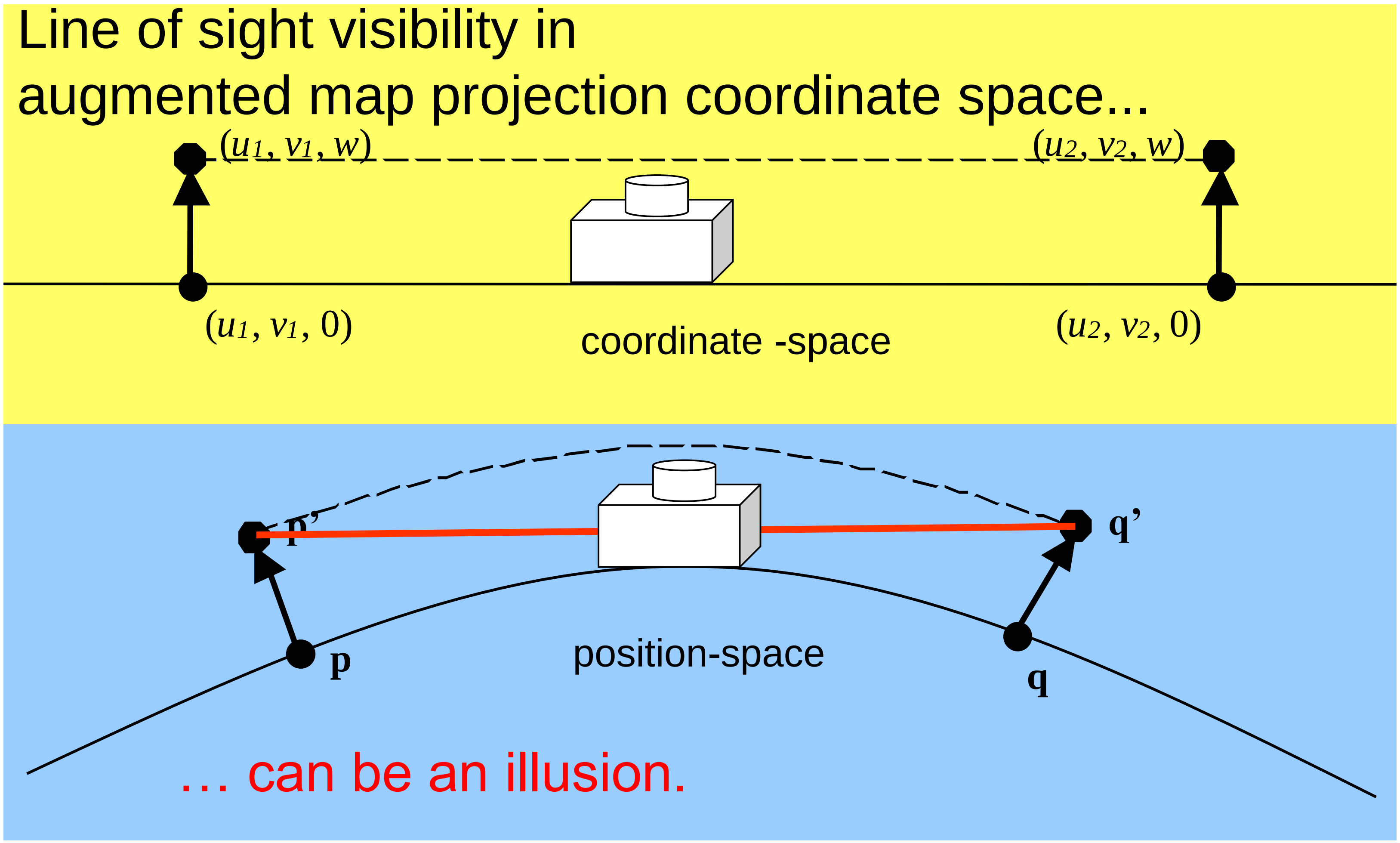
Augmented
MP coordinate (u, v, h) ⟷ (λ, ϕ, h) 3D geodetic coordinate



Caution!

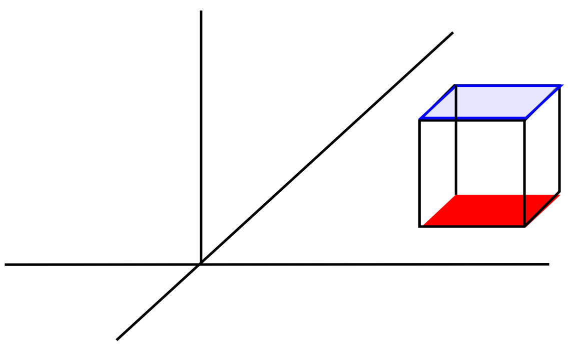
Augmented map geometry in coordinate-space must not be confused with position-space geometry

Augmented map projections

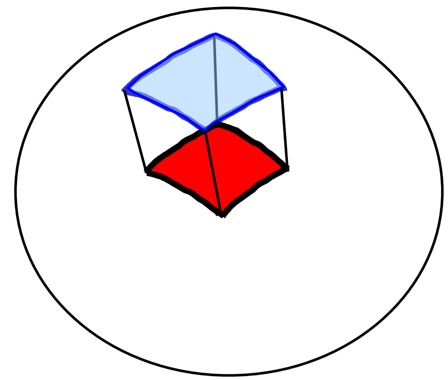


Geometry: coordinate-space *vs*. position-space

A cube in an augmented map projection coordinate-space ...



... is not a cube position-space.



- Every point on the base (red) is on the plane.
- Interior angles of the base are 90°.
- All other interior angles are 90°.
- All sides are of the same length.
- The vertical sides are parallel planes.
- The cube is a convex hull.

- Only points in the red region are transformed by the map projection.
- Every point on the base (red) is on the ellipsoid.
- If the projection is *conformal* interior angles of the base are 90°.
- All other angles are generally not 90°.
- In general none of the sides are equal.
- The vertical lines are not parallel and are not even coplanar.
- The 3D volume is no longer convex.

Standard coordinate systems

3D CS

Euclidean 3D
Lococentric Euclidean 3D
Spherical
Lococentric spherical
Azimuthal spherical
Lococentric azimuthal spherical
Geodetic 3D
Cylindrical
Lococentric cylindrical

Surface CS

Surface geodetic
Lococentric surface Euclidean
Lococentric surface azimuthal
Lococentric surface polar

Plus those that will be registered with the ISO standard.

Map projection CS

Euclidean 1D
Mercator
Oblique Mercator
Transverse Mercator
Lambert conformal conic
Polar stereographic
Equidistant cylindrical

2D CS

Euclidean 2D
Lococentric Euclidean 2D
Azimuthal
Lococentric azimuthal
Polar
Lococentric polar

1D CS:

Euclidean 1D

Surface CS induced on a coordinate surface

- Geodetic 3D
  - Surface geodetic induced on $h = 0$ surface.
- Augmented Map projection
  - Map projection induced on $h = 0$ surface.
- Cylindrical
  - Surface Polar induced on $z = 0$ surface.

Temporal CS

An *integrated temporal coordinate system*

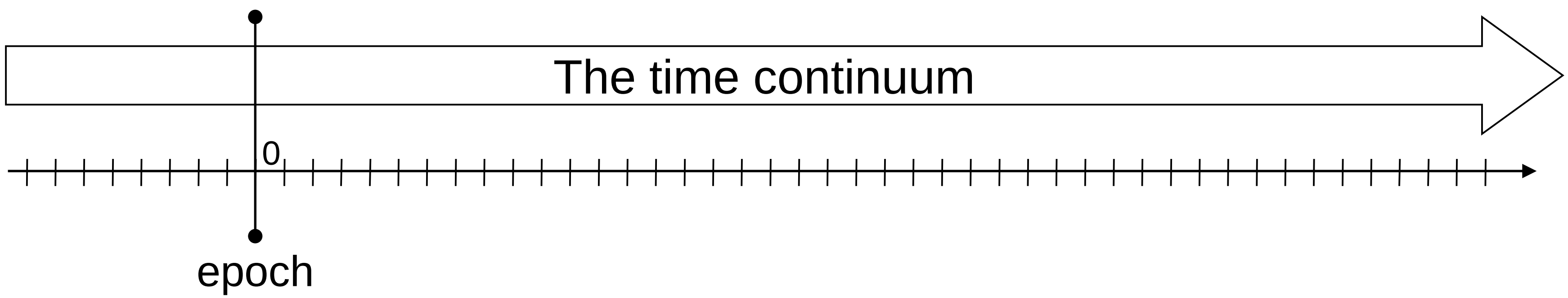
is a Euclidean 1D CS based on a unit of duration

that is derived from a physical phenomenon.

Fixing an origin (called the *epoch*) and then continuously

accumulating units of the duration specifies an

integrated temporal coordinate system.



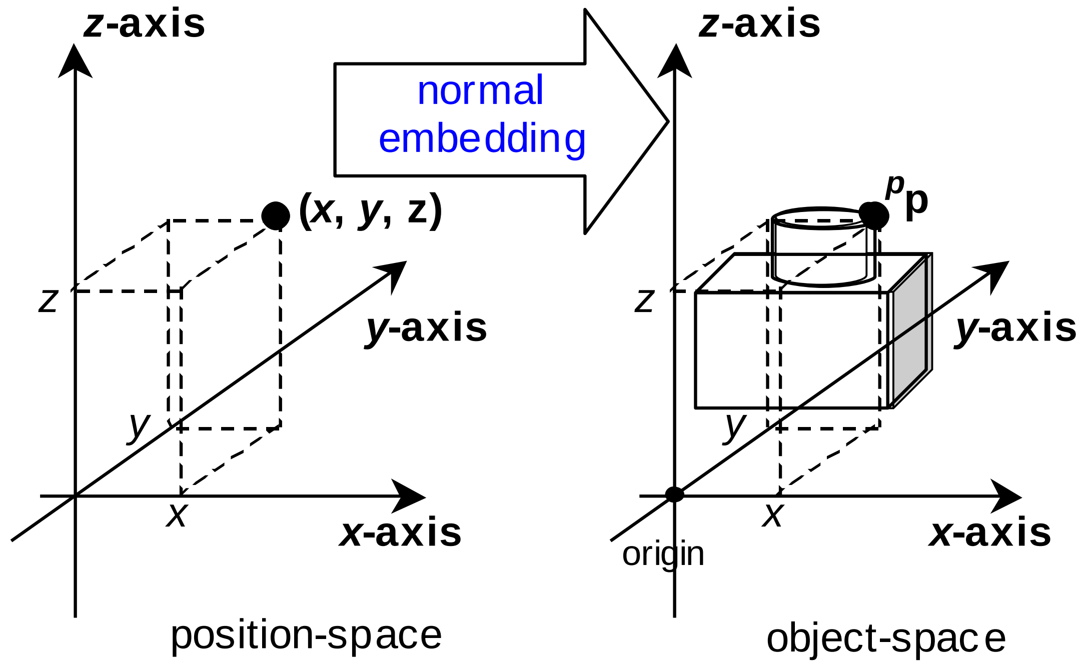
Spatial coordinate systems

Abstract CS + position-space normal embedding

= Spatial CS

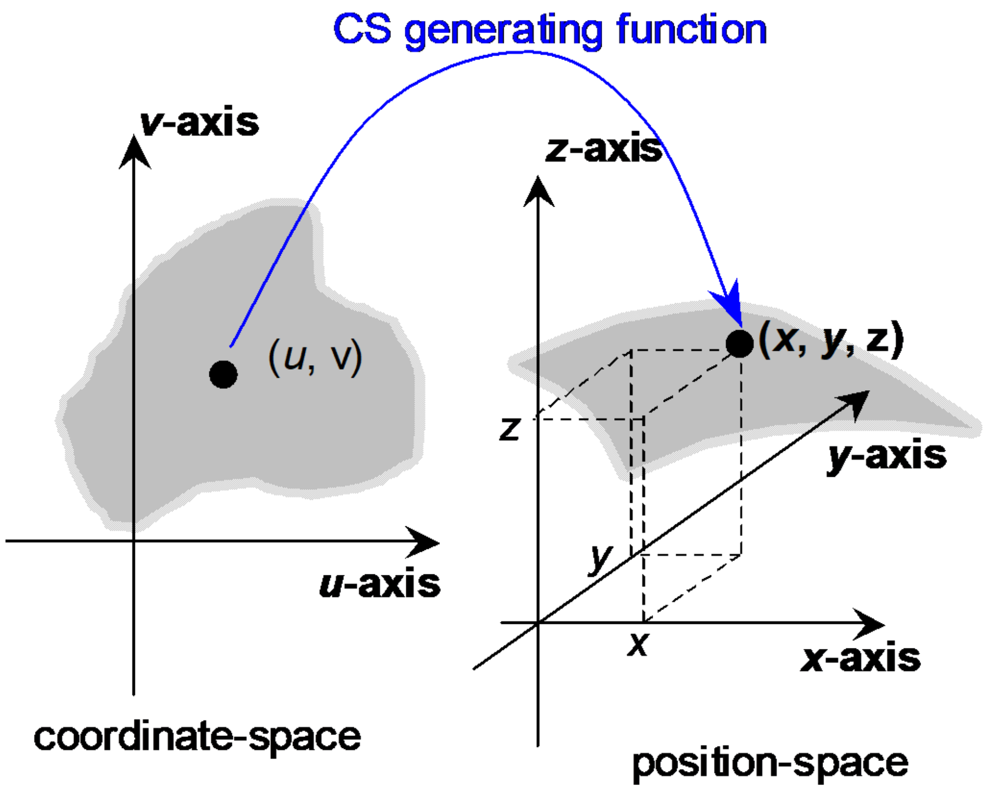
Spatial coordinate systems

3D case



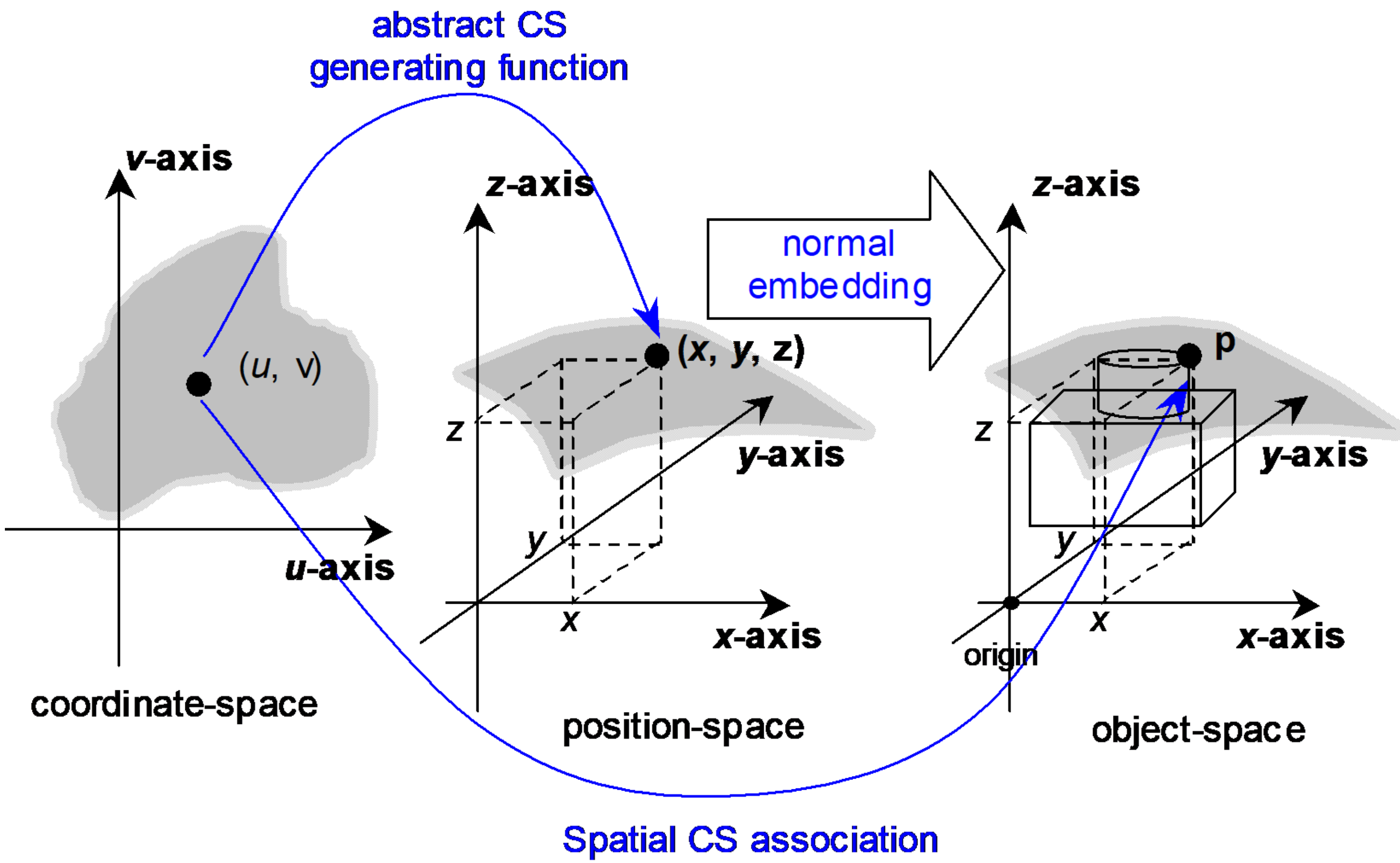
Spatial coordinate systems

3D case



Spatial coordinate systems

3D case



SRM concepts

- Spatial objects and object-space
- Position-space and normal embeddings
- Reference datums
- Object reference models
- Coordinate systems
  - Abstract coordinate systems
    - localization
    - Induced surface coordinate systems
    - map projections
  - Temporal coordinate systems
  - Spatial coordinate systems
- **Spatial reference frames**
- Vertical offset surfaces
- Spatial operations

Spatial reference frame (SRF)

Concept

When a CS is bound to the embedding specified by an ORM, the result is called a Spatial Reference Frame.

Spatial reference frame

- Specification
  - An ORM with a compatible CS binding
    - Example:
      Ellipsoid based CS is compatible with Ellipsoidal ORM only.
    - CS binding:
      ORM determines (some) CS parameters.
    - Example: geodetic parameters *a*, *b* are set to oblate ellipsoid major and minor semi-axis values
  - Adds additional domain CS restrictions
    - Example: Restrict to a specific region
      - Optionally specify a coordinate ***valid region***
      - Optionally specify a coordinate ***extended valid region***
  - Values for remaining CS parameters
    - If any CS parameters are not bound by the ORM

Celestiodetic SRF

Example:

geodetic 3D CS

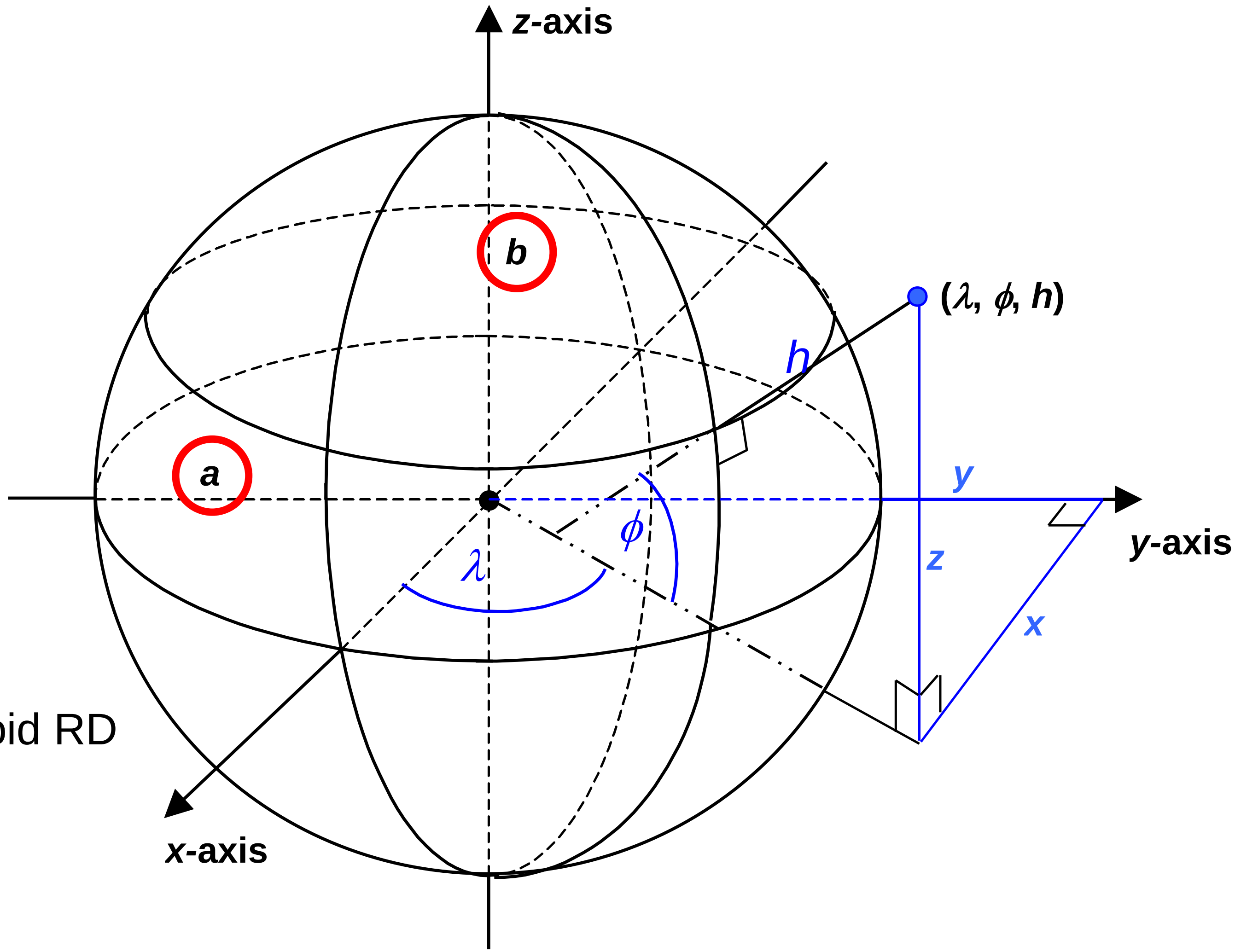
+

Parameter binding to the ellipsoid of a ellipsoidal ORM

$\Rightarrow$

celestiodetic SRF

Celestiodetic SRF



Geodetic CS parameters:
a, *b*

Bound to ORM oblate ellipsoid RD parameters:
a, *b*

Local tangent space Euclidean SRF

Example:

Lococentric Euclidean 3D CS

+

Parameter binding to the ellipsoid of a ellipsoidal ORM

$\Rightarrow$

Local tangent space Euclidean SRF

Local tangent space Euclidean SRF

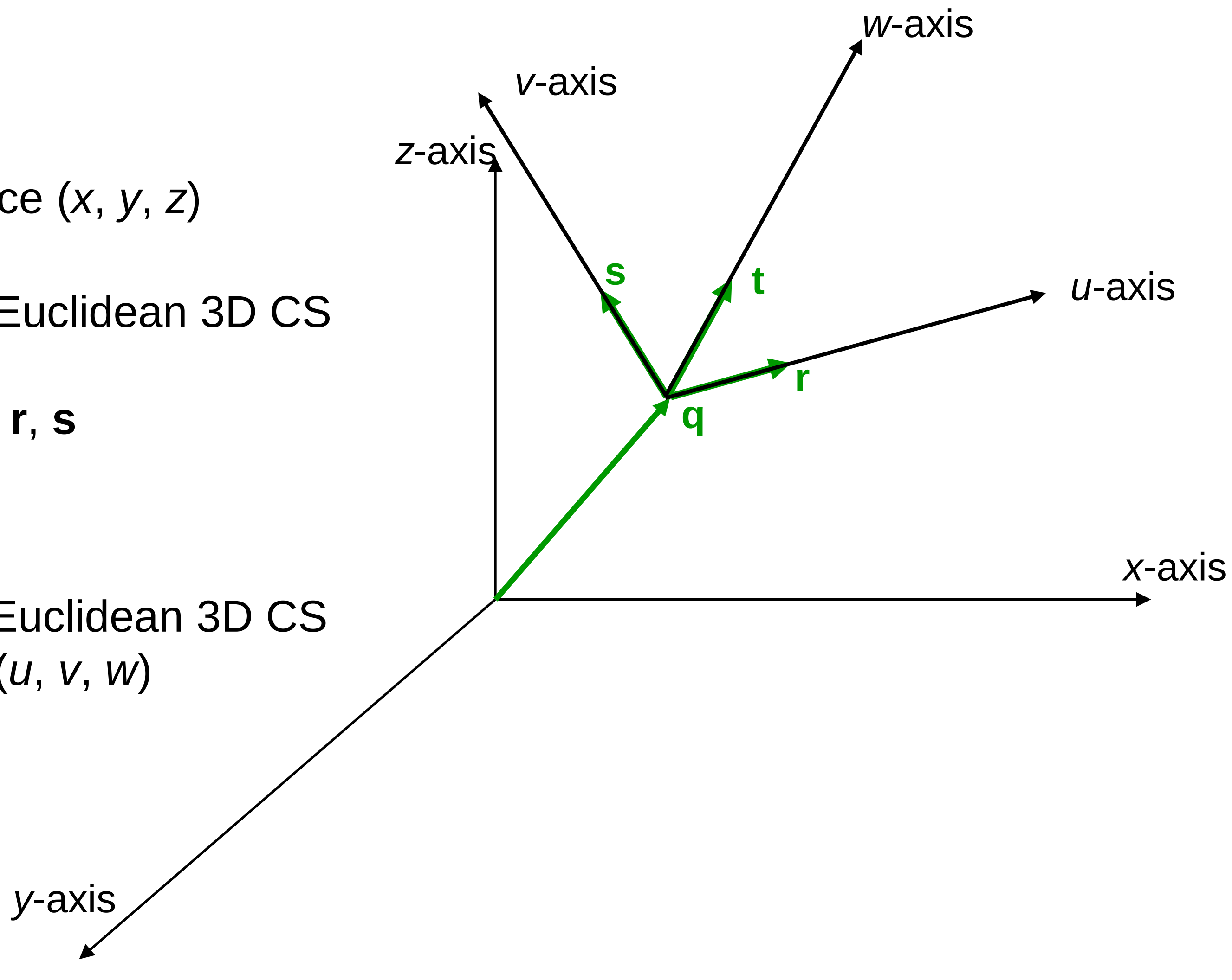
position-space (x, y, z)

Lococentric Euclidean 3D CS parameters:

vectors **q**, **r**, **s**

(**t** = **r** x **s**)

Lococentric Euclidean 3D CS coordinate: (u, v, w)

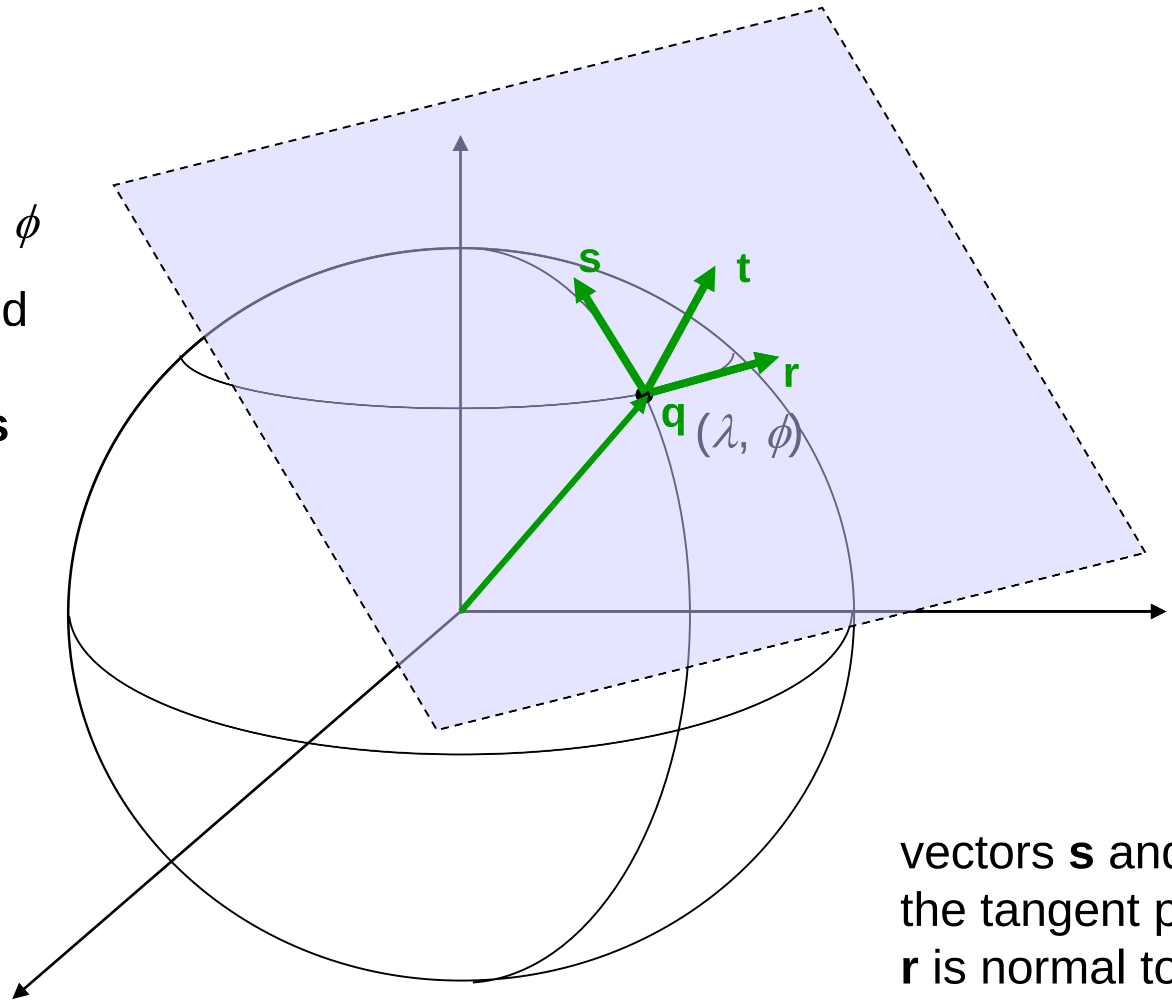


Local tangent space Euclidean SRF

ellipsoid ORM

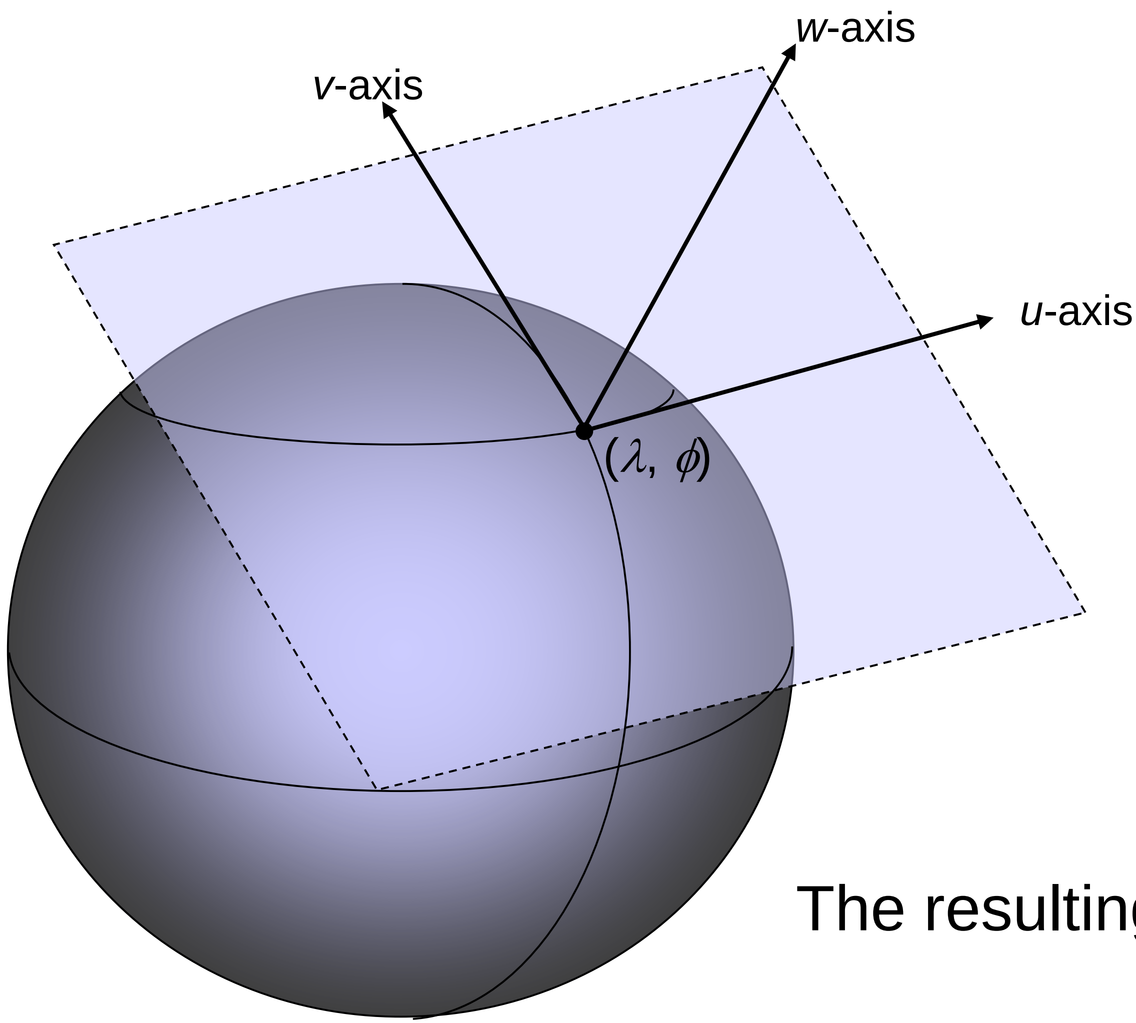
SRF
parameters: λ, ϕ

Use ORM to bind
CS parameters:
vectors **q**, **r**, **s**



vectors **s** and **t**, lie in
the tangent plane
r is normal to the surface

Local tangent space Euclidean SRF

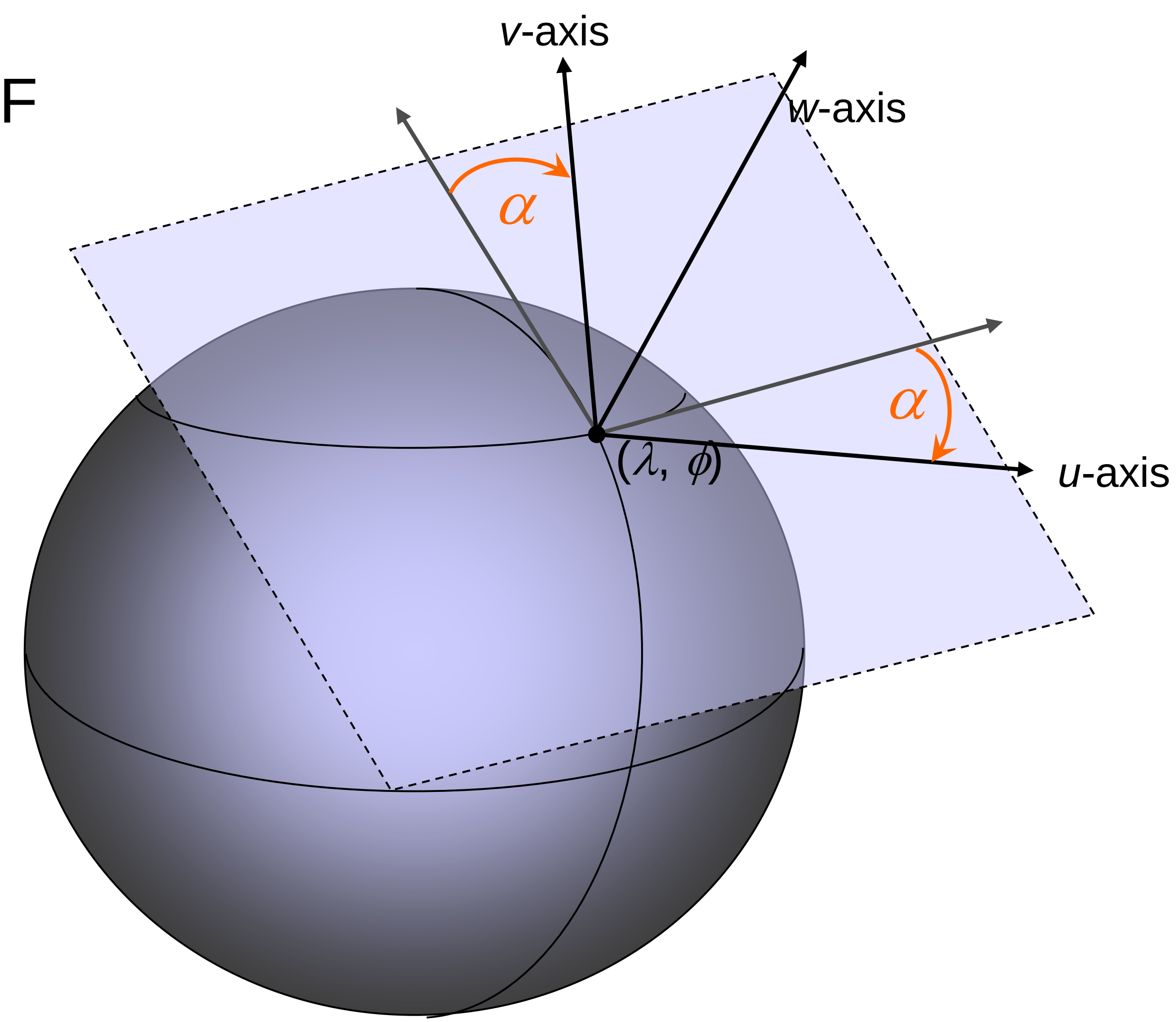


The resulting SRF

Local tangent space Euclidean SRF

Additional SRF parameters:

α: azimuth

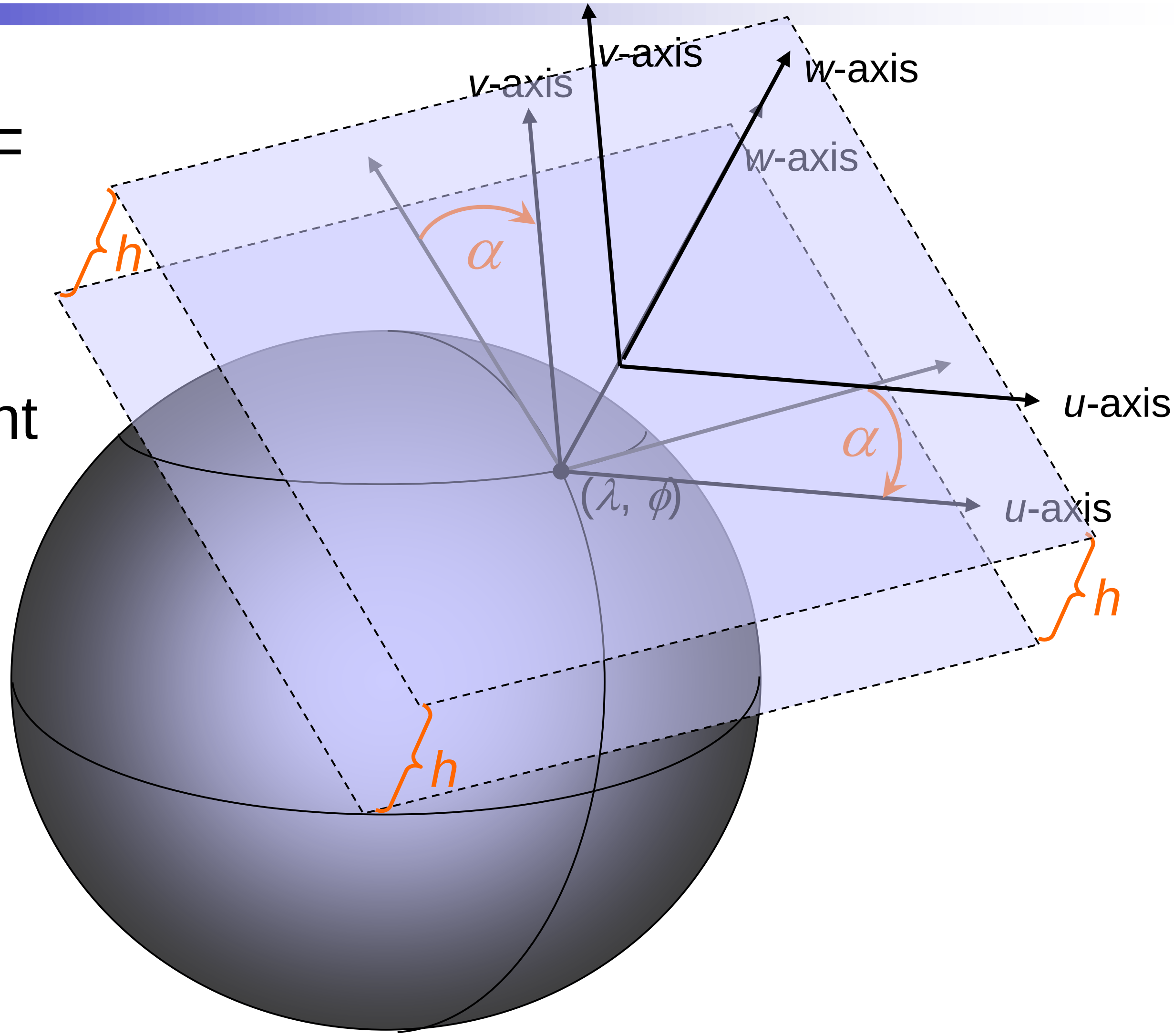


Local tangent space Euclidean SRF

Additional SRF parameters:

α: azimuth

h_0: offset height



Local tangent space Euclidean SRF

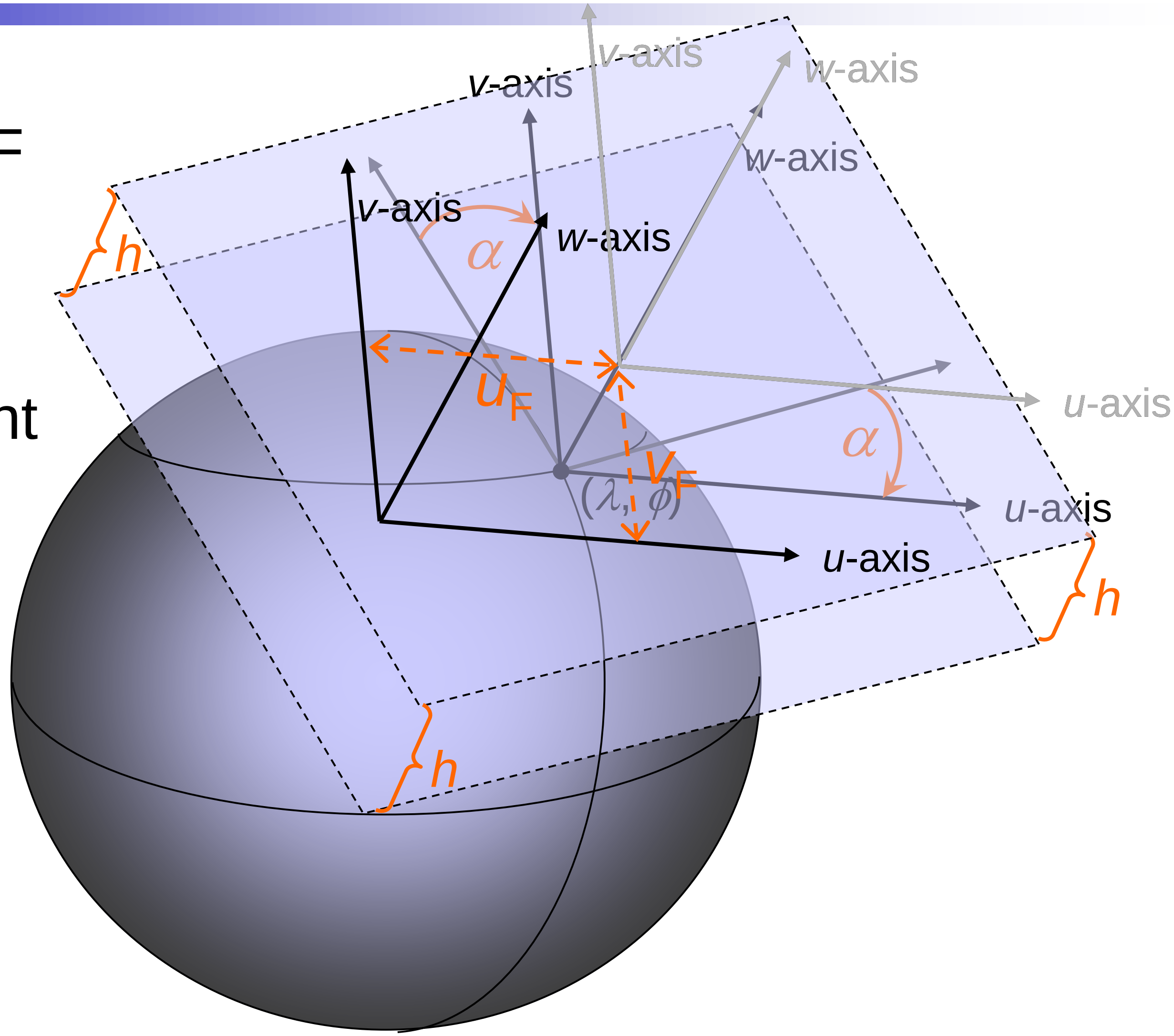
Additional SRF parameters:

α: azimuth

h_0: offset height

u_F: false easting

v_F: false northing



Local tangent space Euclidean SRF

Algebraic specification of the parameter binding

$$\mathbf{r} = \mathbf{T}\begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}$$

$$\mathbf{s} = \mathbf{T}\begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix}$$

$$\mathbf{q} = \mathbf{q}_0 - u_{\mathrm{F}}\mathbf{r} - v_{\mathrm{F}}\mathbf{s}$$

Where:

$$\mathbf{q}_0 = \begin{pmatrix} \left(R_{\mathbf{N}}(\phi) + h_0\right)\cos\phi\cos\lambda \\ \left(R_{\mathbf{N}}(\phi) + h_0\right)\cos\phi\sin\lambda \\ \left(\dfrac{b^2}{a^2}R_{\mathbf{N}}(\phi) + h_0\right)\sin\phi \end{pmatrix}$$

a and b match the

oblate ellipsiod (or sphere) RD values, and

$$\mathbf{T} = \begin{pmatrix} -\sin\lambda & -\cos\lambda\sin\phi & \cos\lambda\cos\phi \\ \cos\lambda & -\sin\lambda\sin\phi & \sin\lambda\cos\phi \\ 0 & \cos\phi & \sin\phi \end{pmatrix}\begin{pmatrix} \cos\alpha & \sin\alpha & 0 \\ -\sin\alpha & \cos\alpha & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

Canonical
Local tangent space Euclidean SRF

Given <u>any</u> SRF based on an oblate ellipsoid ORM (or sphere ORM), and a point (λ_o, ϕ_o) on the ellipsoid, the ***Canonical Local tangent space Euclidean (CLTSE) SRF*** at (λ_o, ϕ_o) is the Local tangent space Euclidean SRF with parameters:

$\lambda = \lambda_o$
$\phi = \phi_o$
$\alpha = 0$
$h_0 = 0$
$u_F = v_F = 0$

CLTSE SRFs are useful for defining a vector **direction** in an SRF with a curvilinear CS.

CLTSE SRFs also useful in instancing an abstract model at a point in an SRF. (E.g.: SEDRIS DRM Geometry/Feature Model Instance.)

SRF templates

- SRF Template (SRFT)
  - ORM constraints and CS parameters binding.
    - Unbound CS parameters (if any) become SRFT parameters.

- SRF instance of a template
  - ORM specified, and
  - SRFT parameters are specified.
  - All standardized SRF in the SRM are SRFT instances.

- SRF sets
  - A fixed (standard/registered) indexed set of SRFT instances for a given ORM.
  - Examples : UTM - zone code and hemisphere
GCS - index code
PS - polar aspect code

SRFTs for 3D object-space

Celestiocentric
Local space rectangular 3D
Celestiodetic
Local tangent space Euclidean
Local tangent space azimuthal spherical
Local tangent space cylindrical
Celestiomagnetic
Equatorial inertial
Solar ecliptic
Solar equatorial
Solar magnetospheric
Solar magnetic
Heliospheric Aries ecliptic
Heliospheric Earth ecliptic
Heliospheric Earth equatorial

(augmented) Map projections

Mercator
Oblique Mercator
Transverse Mercator
Lambert conformal conic
Polar stereographic
Equidistant cylindrical

Plus those that will be registered with the ISO standard.

SRFTs for 3D object-space

Surface CS SRFs induced by 3D CS SRFs

3D CS SRF	*(induced) surface CS SRF*
Celestiodetic	surface Celestiodetic
Local tangent space Euclidean	Local tangent plane Euclidean
Local tangent space azimuthal spherical	Local tangent plane azimuthal
Local tangent space cylindrical	Local tangent plane polar

(augmented) Map projection SRF	*(surface) Map projection SRF*
Mercator	Mercator
Oblique Mercator	Oblique Mercator
Transverse Mercator	Transverse Mercator
Lambert conformal conic	Lambert conformal conic
Polar stereographic	Polar stereographic
Equidistant cylindrical	Equidistant cylindrical

Compare with ISO 19111 concept of a compound CRS

SRF example

Using Lambert conformal conic SRFT

- ORM
  - North American 1927 (NAD 27)
    - includes: NAD 27 ellipsoid RD (Clark 1866 ellipsoid)
- CS
  - Lambert conformal conic map projection
- SRF parameters (unbound CS parameters)
    - longitude of origin 97º West
    - latitude of origin 38º North
    - central scale 1
    - false easting 0
    - false northing 0
- Additional CS domain/range constraints:
  - Restricted to CONUS (expressed as min/max values for map coordinates *u* and *v*)

SRFTs for 2D object-space

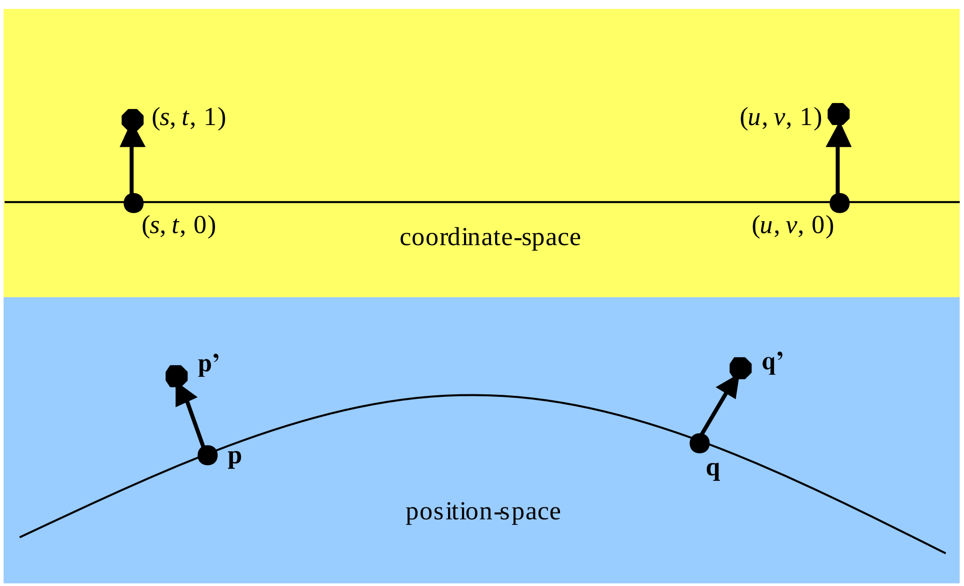
Local space rectangular 2D

Local space azimuthal

Local space polar

Reference vectors

- Represent a **direction** in object space
- Must have reference position
- Defined in SRFs with a <u>linear</u> CS:
  - Celestiocentric, Local Tangent Space Euclidean (LTSE), Local Space Rectangular
- In other cases, defined in Canonical LTSE SRF at the reference location



The "direction" (0,0,1) depends the reference location (**p** or **q**)

Reference vectors

- Reference vectors are used to specify spatial directions.
- In the SEDRIS DRM, reference vectors are restricted to unit vectors associated with a reference position (point).
- Reference vectors are specified in the 3D Orthonormal CS of the CLTSE determined by the reference position.
- Reference vectors and directions in general must be transformed when changing SRFs.

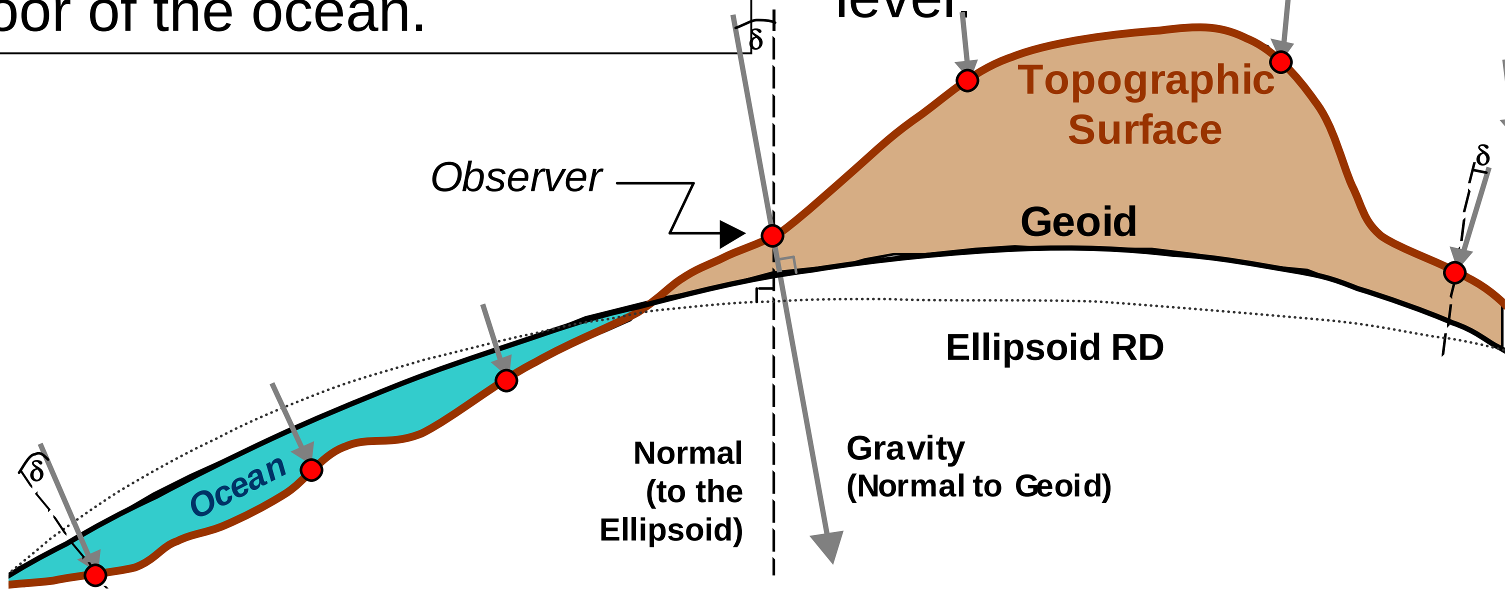
SRM concepts

- Spatial objects and object-space
- Position-space and normal embeddings
- Reference datums
- Object reference models
- Coordinate systems
  - Abstract coordinate systems
    - localization
    - Induced surface coordinate systems
    - map projections
  - Temporal coordinate systems
  - Spatial coordinate systems
- Spatial reference frames
- **Vertical offset surfaces**
- Spatial operations

Earth surfaces

- *Topographic Surface* is the interface between the solid and liquid/gas portions of the Earth. It corresponds to the surface of the land and the floor of the ocean.

- *Geoid Surface* excludes the topographic surface, and therefore generally corresponds with mean sea level.



In the SRM, *there is no Earth*; there are only *models of the Earth*.

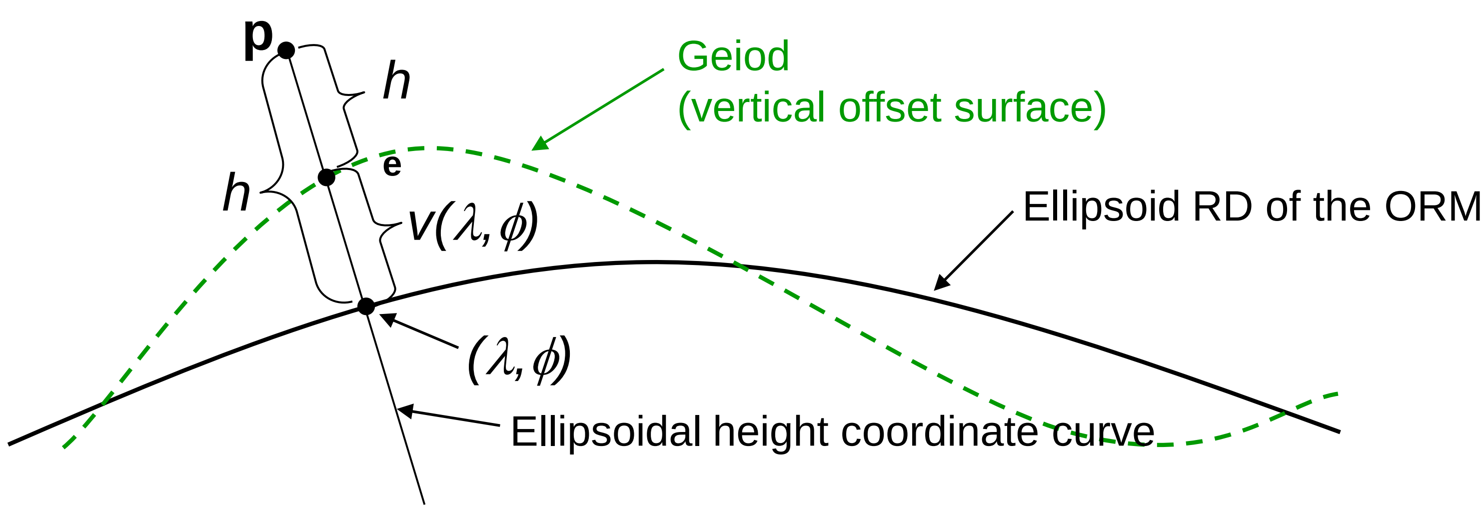
Vertical offset model (VOM)

- A VOM is a smooth surface defined by a function the embedded position-space of an ORM
- A VOM models a reference surface
  - I.e. Mean sea level, gravity equi-potential surface, etc.
- Example:
  - The geiod for WGS 84 ORM

Vertical offset model (VOM)

- Can be used to modify height for a composite SRF:

ellipsoidal height $h \Rightarrow$ elevation $h_e = (h - v(\lambda, \phi))$
where $v(\lambda, \phi)$ is the geiodal separation value



SRM concepts

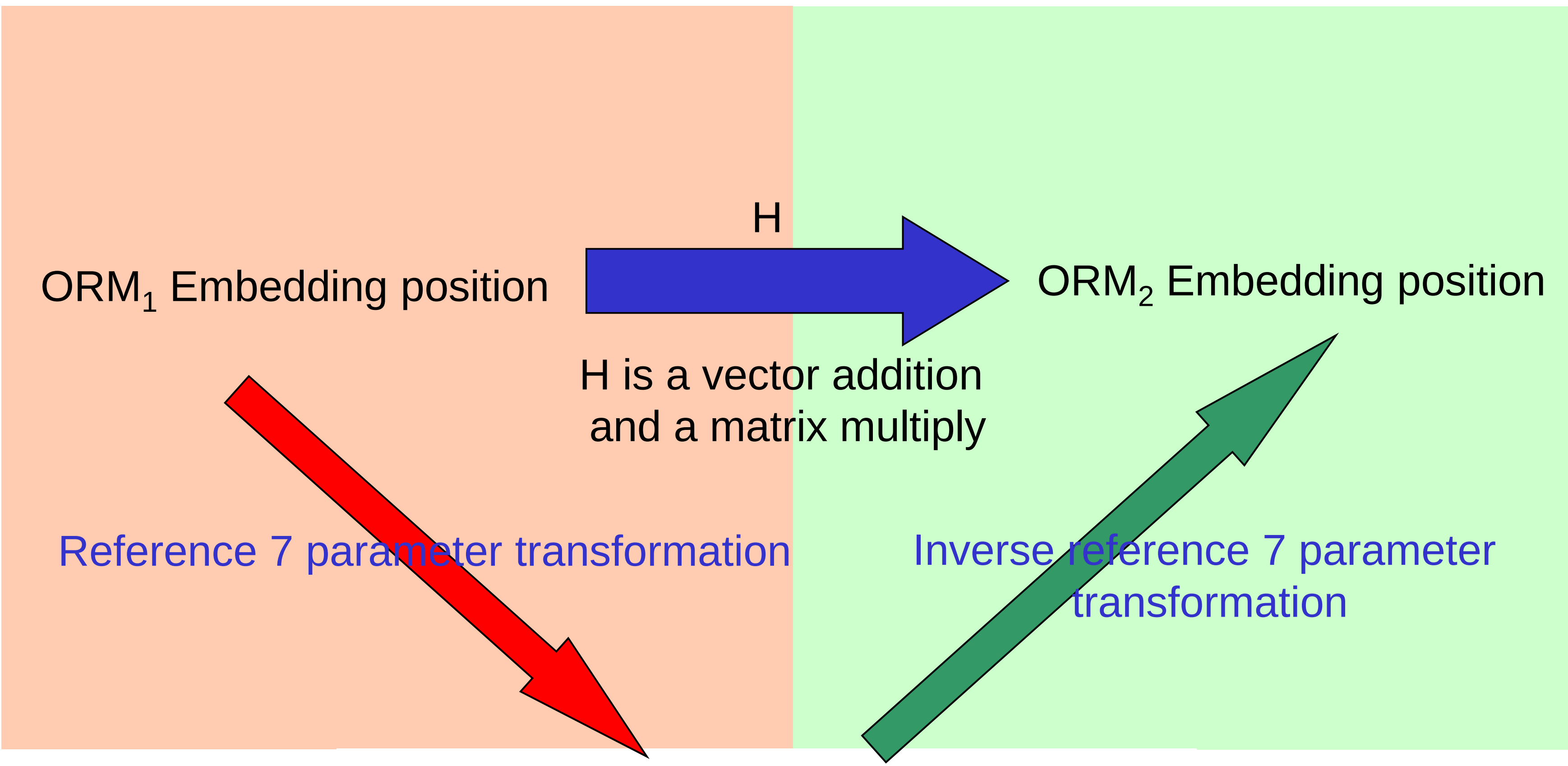
- Spatial objects and object-space
- Position space and normal embeddings
- Reference datums
- Object reference models
- Coordinate systems
  - Abstract coordinate systems
    - localization
    - Induced surface coordinate systems
    - map projections
  - Temporal coordinate systems
  - Spatial coordinate systems
- Spatial reference frames
- Vertical offset surfaces
- **Spatial operations**

Spatial operations

- Change SRF
  - Coordinate representation
    - includes as special cases:
      coordinate transformation
      coordinate conversion
    - coordinate validity
  - Direction representation
  - Change position-space embedding
- Map projection based SRFs
  - map and geodetic azimuth
  - point scale
  - Convergence of the Meridian
- Euclidean and geodesic distance

Change of embeddings

Case: object-fixed ORMs for same object

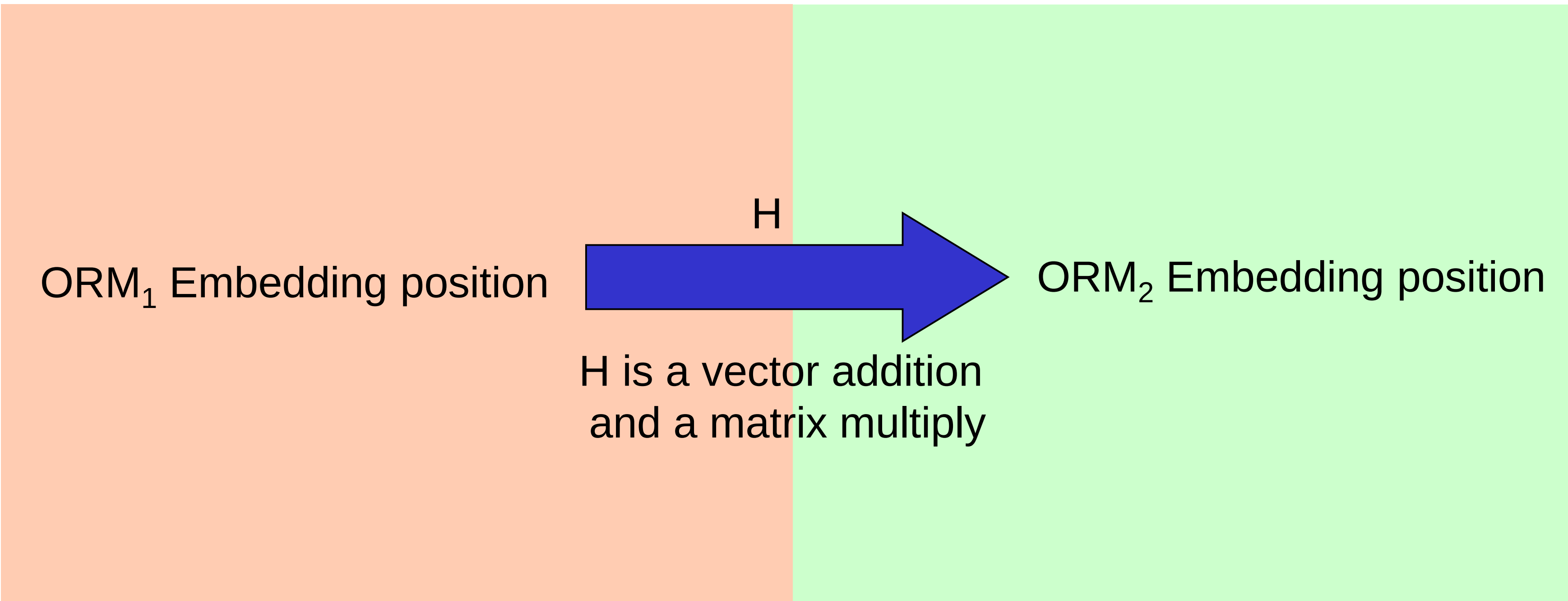


Object reference ORM Embedding position

(Reference ORM for Earth is WGS 84 ORM)

Change of embeddings

Case: dynamic ORMs or two different objects



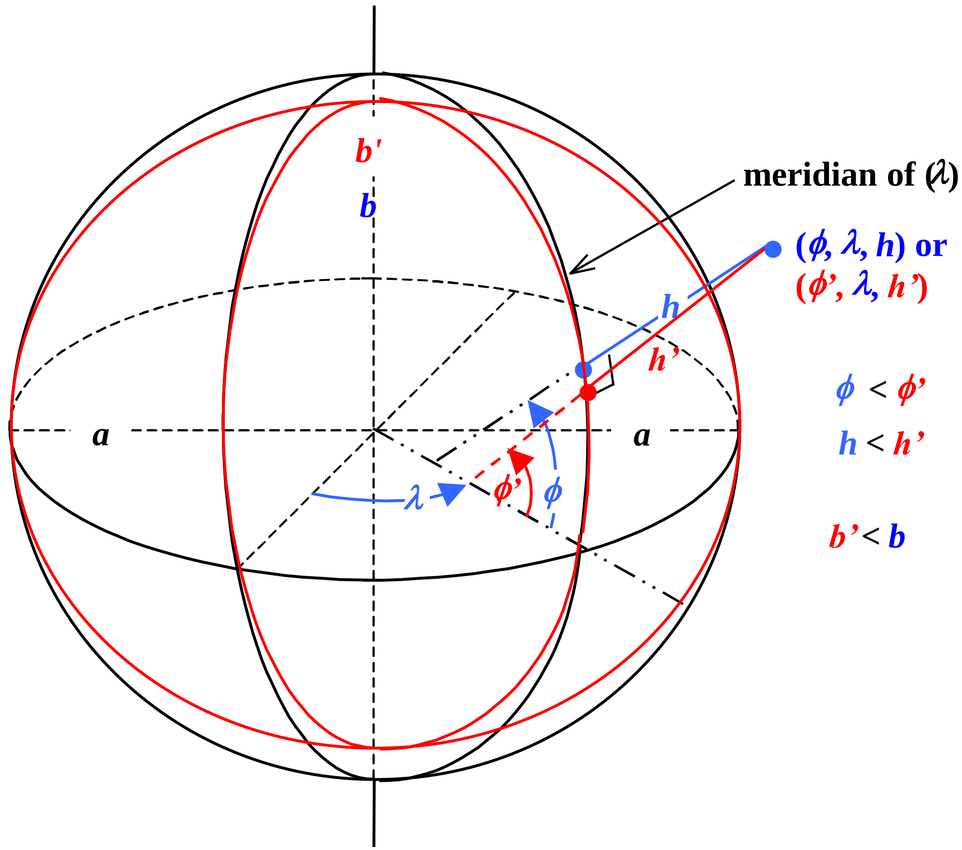
Assuming a fixed time coordinate $t = t_o$ in the dynamic case:

$$\mathbf{H}=\mathbf{H}(t_o)$$

Coordinate transformation

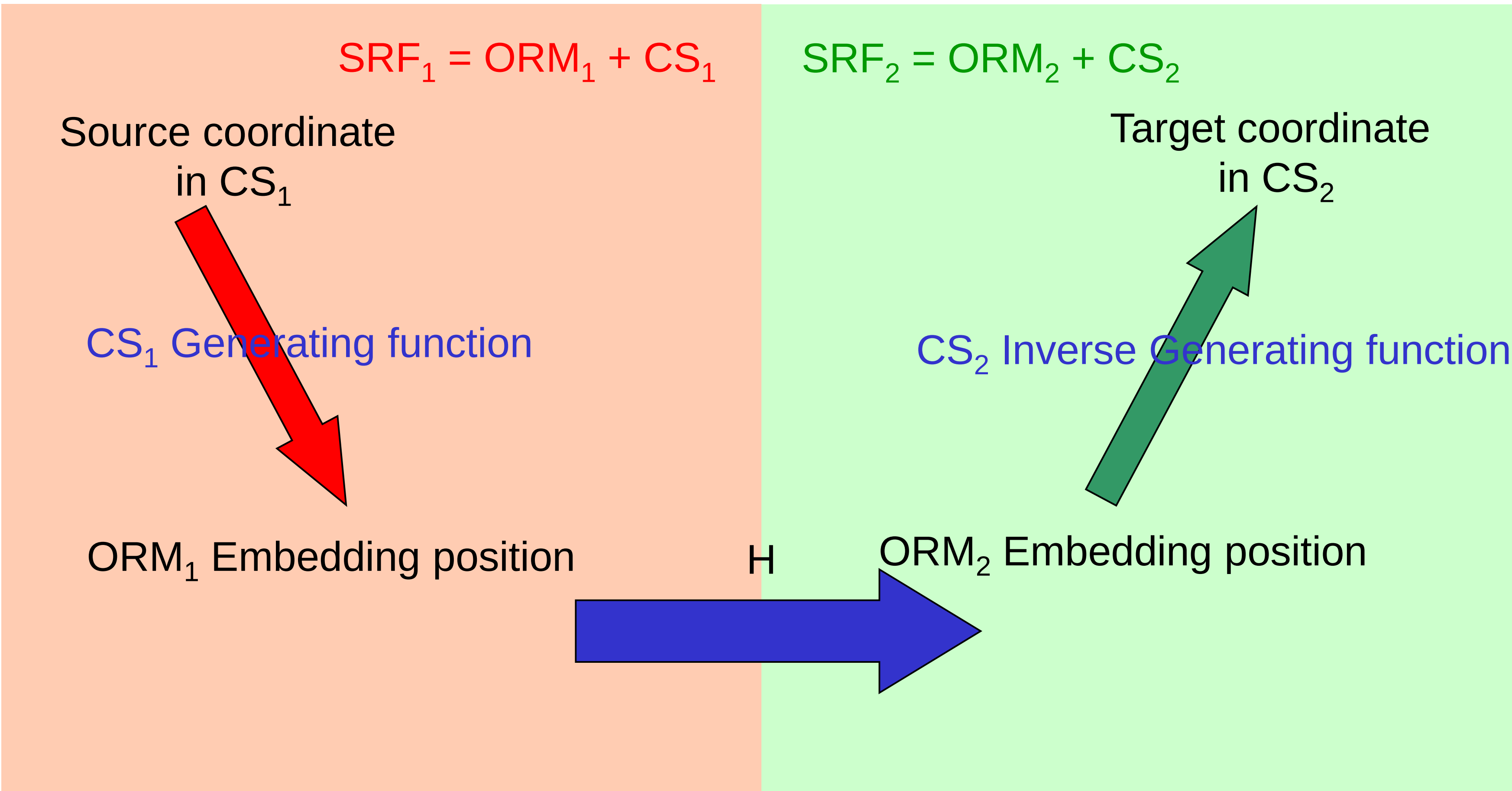
Different ORMs ***(same celestial object)***

Same CS



Change SRF operations

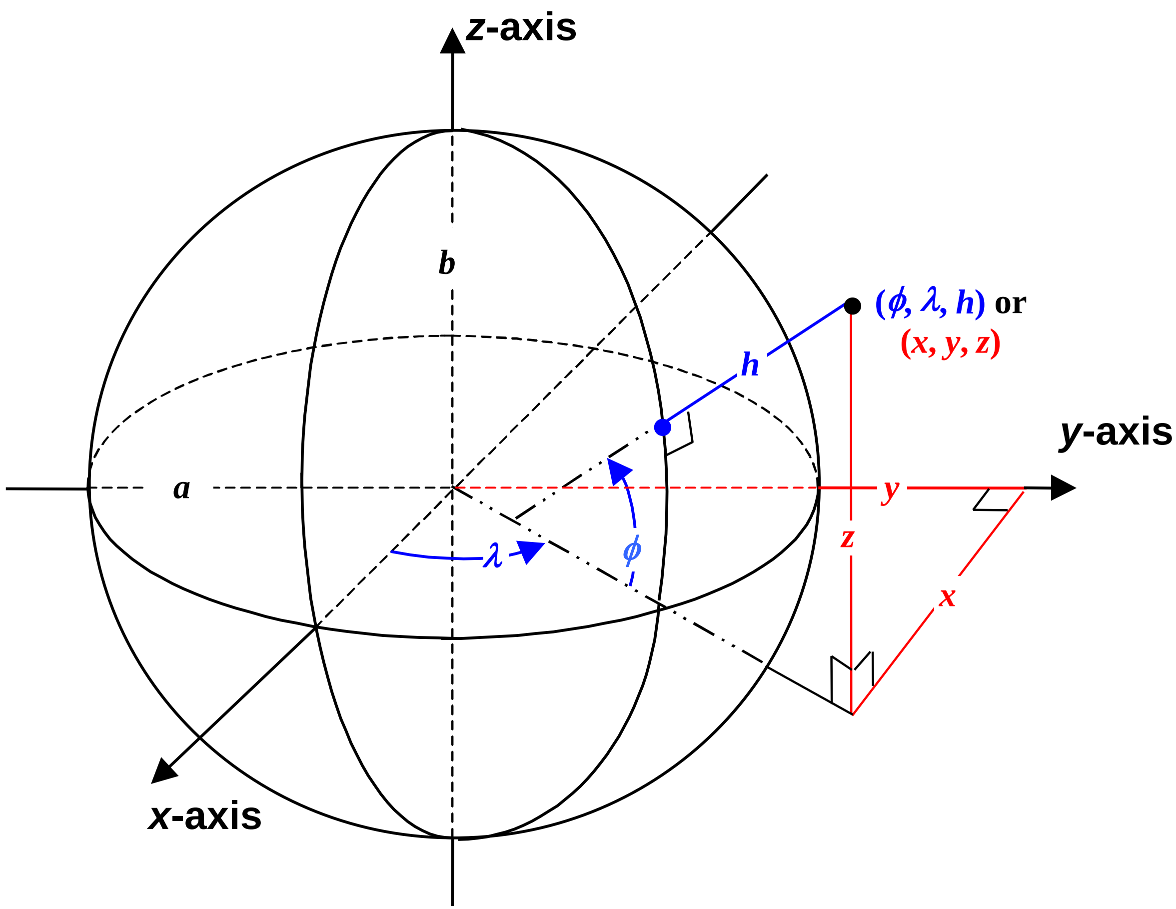
Coordinates



Coordinate conversion

Same ORM

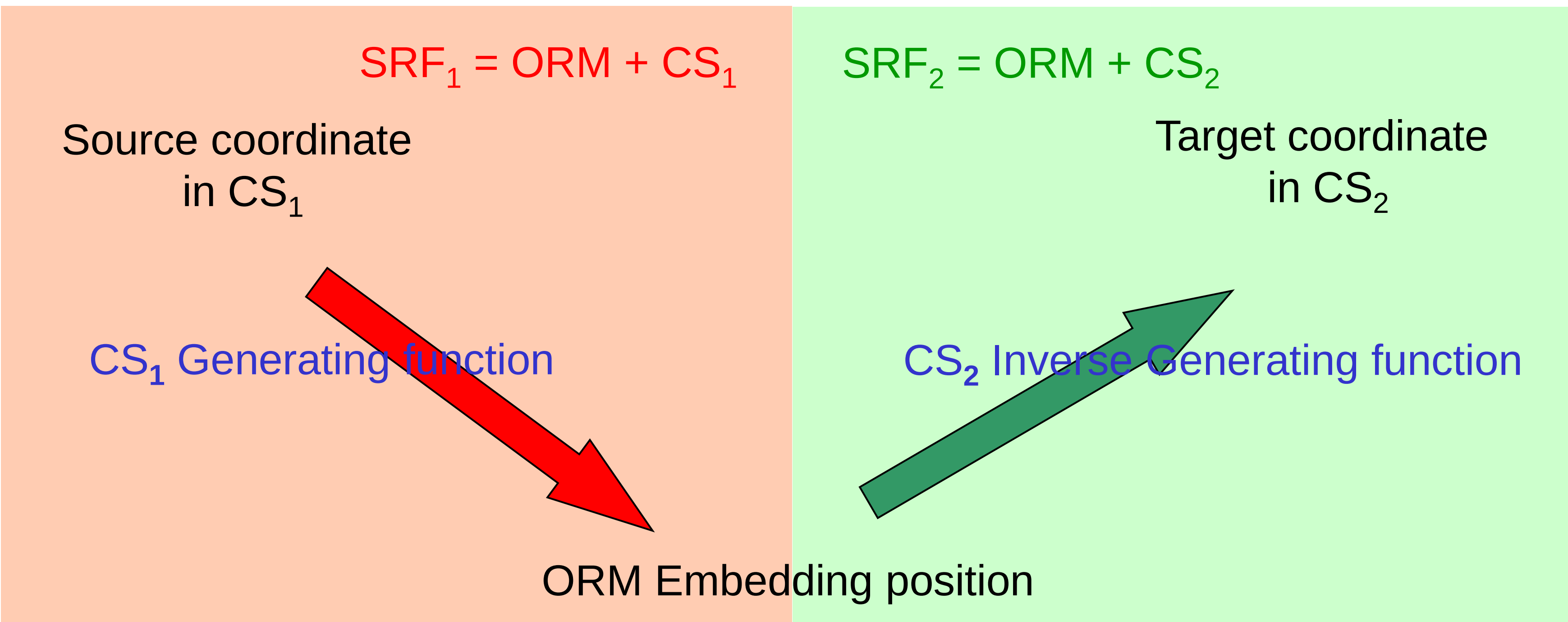
Different CS



Change SRF operations

Special Case: Coordinate Conversion

Both SRFs use the *same* ORM (**H** not needed)



Spatial operations

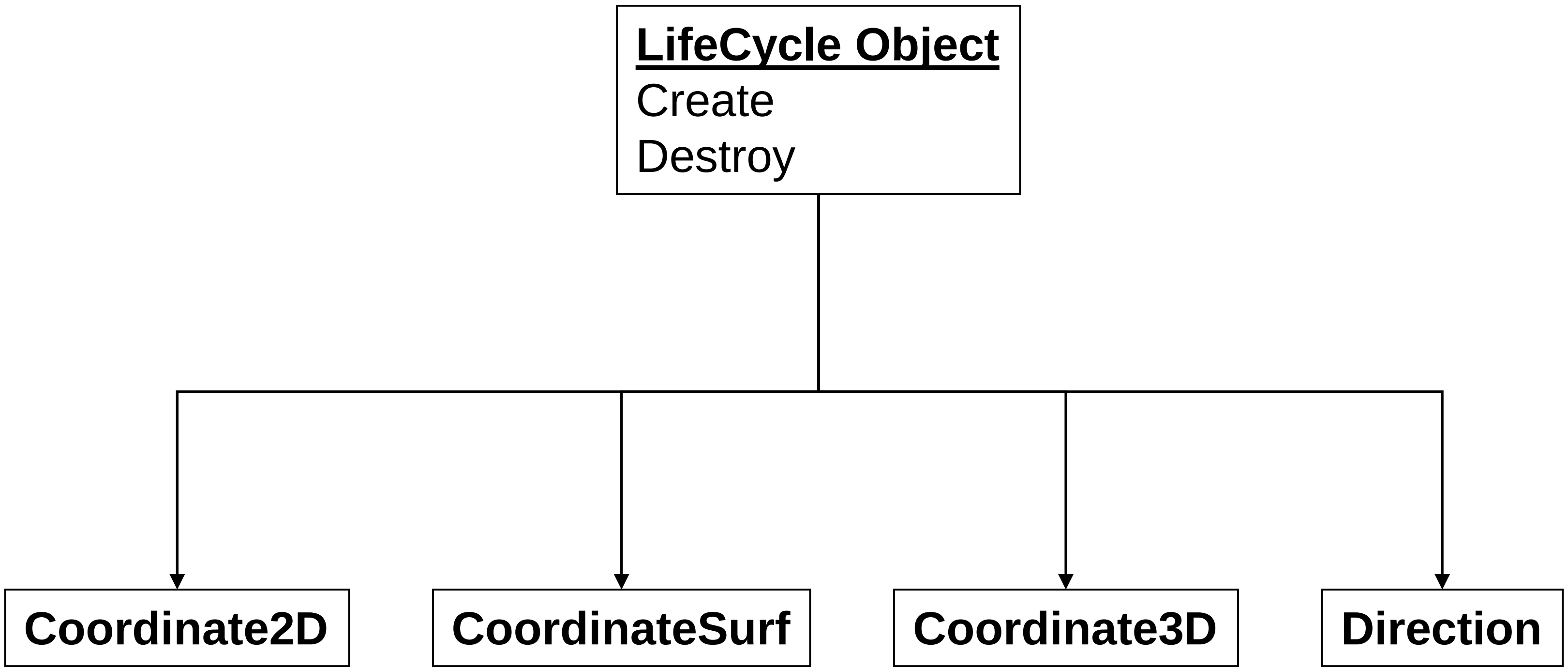
- **Validation**: function for determining coordinate validity after a change SRF operation.
  - *valid* when the coordinate falls within the valid region specification for that SRF;
  - *extended valid* when the coordinate falls outside of the valid region specification for that SRF, but falls within the extended valid region specification for that SRF;
  - *defined* when the coordinate falls within the domain of the CS specified for that SRF; and
  - *invalid* when the coordinate falls outside of the CS domain of the SRF Class of that SRF Class instance.
- **Example** UTM
  - Valid region: $\pm 3°$ of UTM zone central meridian.
  - Extended valid region: $\pm 6°$ of UTM zone central meridian.

Abstract API

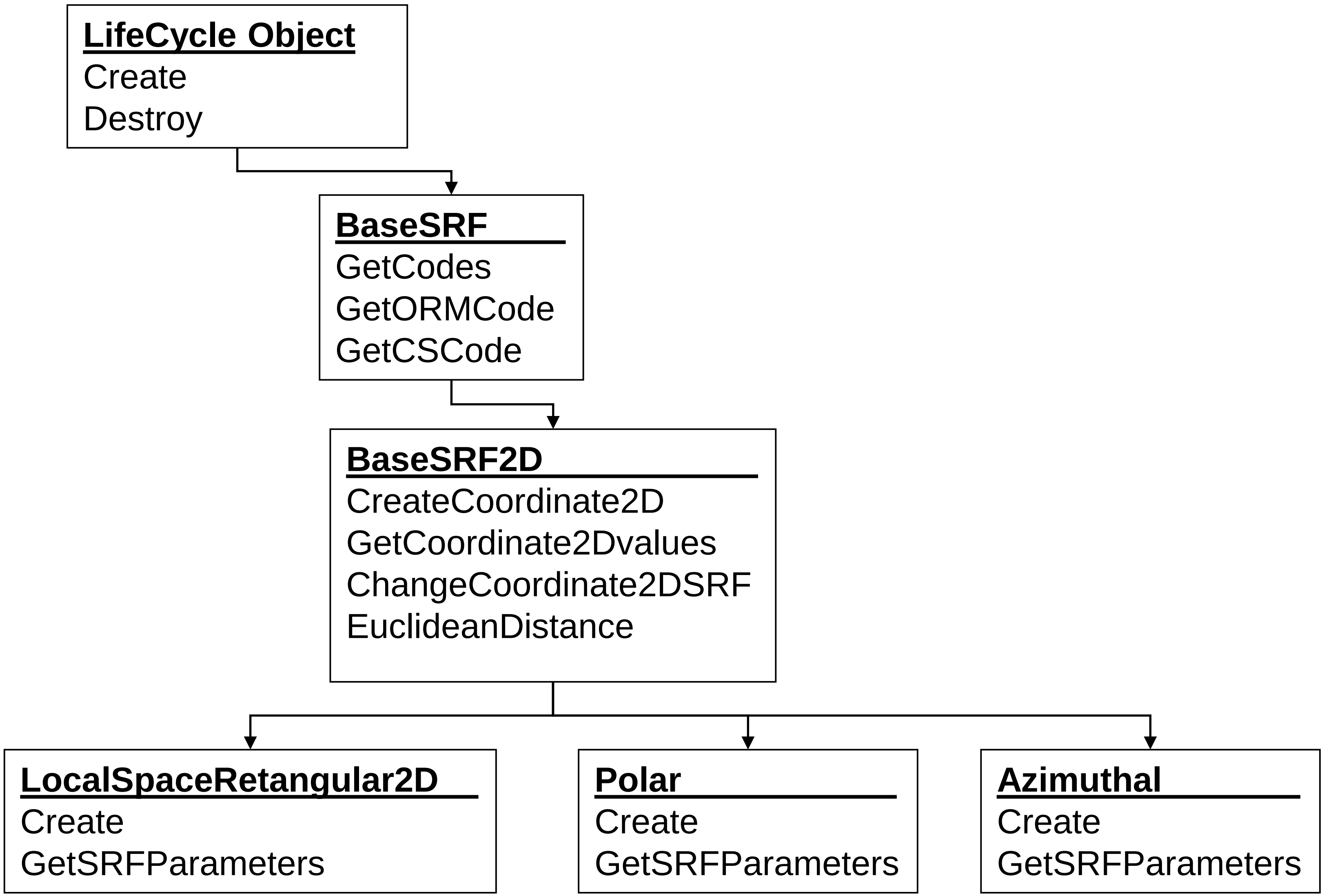
SRM API

- Abstract specification
  - Language bindings are separate documents.
- SRF objects are instances of SRFT classes.
- Most operations are methods on SRF objects.
  - A coordinate object only has meaning in the context of an SRF.
- Hierarchical structure: Methods common to a sub-class of SRFT classes are documented in abstract base classes.

SRM API object hierarchy (1 of 6)



SRM API object hierarchy (2 of 6)



SRM API object hierarchy (3 of 6)

LifeCycle Object

Create
Destroy

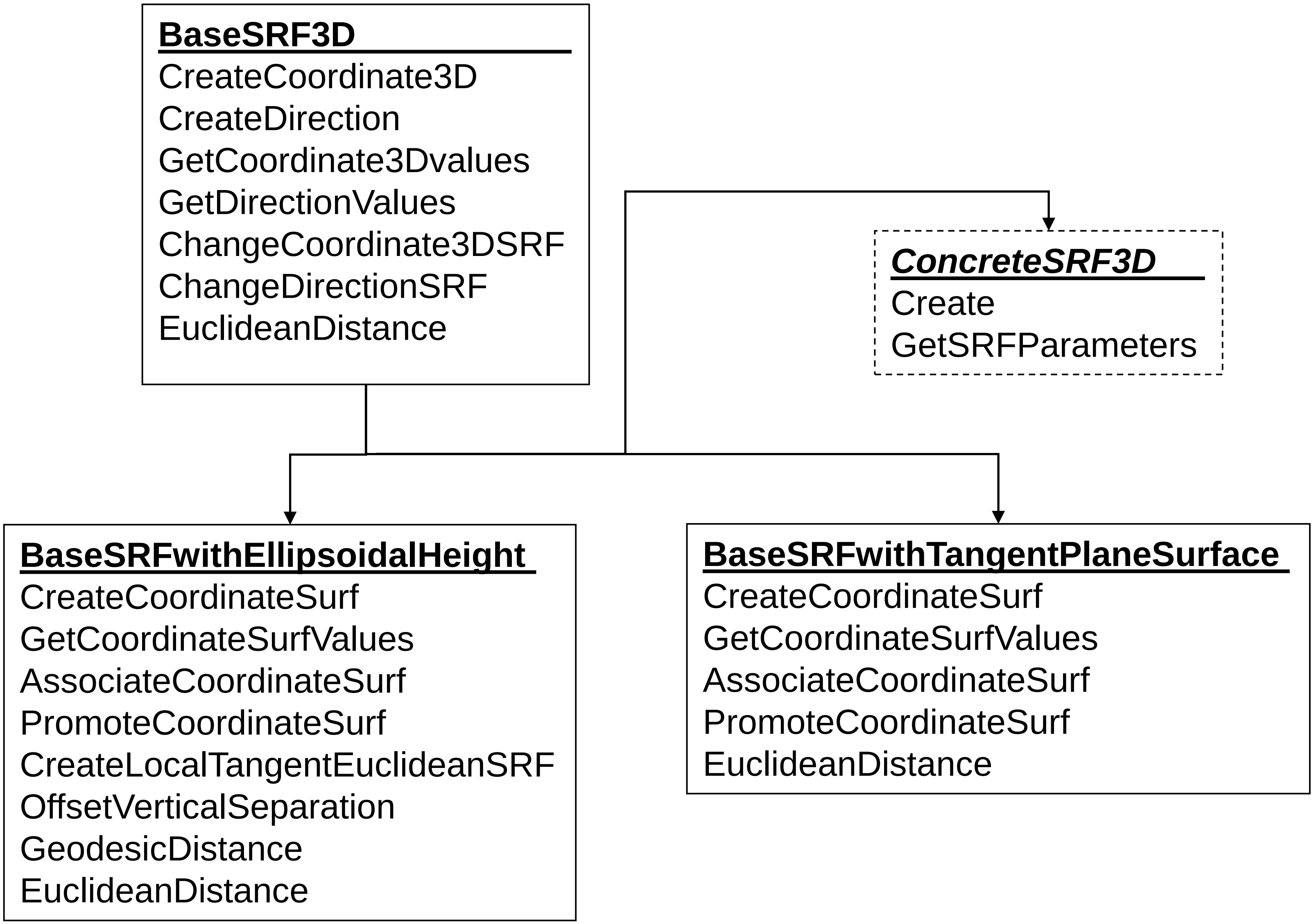
BaseSRF

GetCodes
GetORMCode
GetCSCode

BaseSRF3D

CreateCoordinate3D
CreateDirection
GetCoordinate3Dvalues
GetDirectionValues
ChangeCoordinate3DSRF
ChangeDirectionSRF
EuclideanDistance

SRM API object hierarchy (4 of 6)



SRM API object hierarchy (5 of 6)

BaseSRFwithEllipsoidalHeight
CreateCoordinateSurf
GetCoordinateSurfValues
AssociateCoordinateSurf
PromoteCoordinateSurf
CreateLocalTangentPlaneSRF
OffsetVerticalSeparation
GeodesicDistance
EuclideanDistance

Celestiodetic
Create
GetSRFParameters

BaseMapProjectionSRF
ConvergenceOfTheMeridian
PointScale
MapAzimuth

ConcreteMapProjection
Create
GetSRFParameters

SRM API object hierarchy (6 of 6)

BaseSRFwithTangentPlaneSurface
CreateCoordinateSurf
GetCoordinateSurfValues
AssociateCoordinateSurf
PromoteCoordinateSurf
EuclideanDistance

LocalTangentEuclidean
Create
GetSRFParameters

LocalTangentAzimuthalSpherical
Create
GetSRFParameters

LocalTangentCylindrical
Create
GetSRFParameters

SRM API non-object functions

CreateStandardSRF3D

Input:	SRF_Code
Output:	SRF3D

CreateSRFSetMember

Inputs:	SRFS_Code
	SRFS_Member_Code
	ORM_Code
Output:	SRF3D

Abstract API examples

Example 1: Find the Euclidean distance between two locations.

--Note: Label in italics denotes a symbolic constant for this example --

Celestiodetic method Create
(Input: *ORM_N_AM_1983_CONUS*; Output: srf)

srf method CreateCoordinate3D
(Inputs -77º(π/180º), +38º(π/180º), 0; Output: coordinate1)

srf method CreateCoordinate3D
(Inputs +3º(π/180º), +49º(π/180º), 0; Output: coordinate2)

srf method EuclideanDistance
(Inputs coordinate1, coordinate2, 0; Output: distance)

-- use distance result --

coordinate1 method destroy
coordinate2 method destroy
srf method destroy

Abstract API examples

Example 2: Change SRF representation of a location from UTM to Celestiocentric

--Note: Labels in italics denote symbolic constants for this example --

Function CreateSRFSetMember
(Input: *SRFS_UNIVERSAL_TRANSVERSE_MERCATOR,*
ORM_N_AM_1983_CONUS,
ZONE_23_NORTHERN_HEMISPHERE, Output: source_srf)

Function CreateCoordinate3D
(Inputs 120, 400, 0, Output: source_coordinate)

Function CreateStandardSRF
(Input: *SRF_GEOCENTRIC_EARTH_1984*, Output: target_srf)

srf method ChangeCoordinate3DSRF
(Inputs source_srf, source_coordinate, Output: target_coordinate)

-- use target_coordinate result --

source_coordinate method destroy
target_coordinate method destroy
source_srf method destroy
target_srf method destroy

SRM API Implementation

STC Tutorial

4 - 6 PM, Tuesday, January 6

SRM for Programmers

Computational considerations

Accuracy

Errors

Testing

Algorithm Design

Error sources in spatial operations software

- There are many possible error sources in development of software for coordinate transformations.
  - Truncation errors are due to the use of a finite number of terms in an infinite series.
  - Approximation error is due to approximating one function with another (simpler to compute) function.
  - Iteration error is due to the use of a finite number of iterations in an iterative process.
  - Formulation errors are due to the analyst developing the incorrect equations or logic – this includes improper formulations near singular points, improper treatment of signs, incorrect treatment of units and others.
  - Implementation errors are due to improper coding of the correct formulation.
  - Round-off errors are those caused by finite word length computers.

Bounds checking

- Coordinate transformations may only be valid for prescribed regions.
- Both input and output should be checked for validity.
- For many coordinate frameworks, the mathematical formulation may exist everywhere even though the results may be nonsensical for some regions.
- Determining acceptable bounds requires an examination of distortion effects and, in some cases, computational accuracy.
- This is an emerging issue for SRM standardization efforts.
- The problem is multi-faceted:
  – What are the acceptable bounds?
  – For which applications?
  – Who determines them?

Distortion and computation error tradeoff

- The computational complexity and efficiency of series representations depend on the number of terms used.
- For map projection-based SRFs, distortion generally increases away from the origin.
- Computational accuracy may be swamped by errors due to distortion.
- Appropriate tradeoff of these effects are application dependent, and is central to determining validity bounds.
- There is little published information to support a cogent decision on how to set the bounds.

How should accuracy be specified?

- Published standards for mathematical library routines set a precedent for this Issue.
  - *E.g.*, IEEE standards specify a valid operating range and accuracy for each function.
- Examples of what happens when the argument is out of range:
  - A negative argument for a real square root, or a zero input for a logarithm will generate error messages.
  - Too large (or small) an argument may generate a warning or require special handling (overflow-underflow protection).
- Computational performance (efficiency) is not specified, but is market driven.
- Accuracy is specified in decimal digits (or bits of precision).
  - Accuracy is specified for the worst-case application.
  - Nominal applications never come close to needing full accuracy.
  - No options are provided for reduced accuracy – this leads to ad hoc (non-standard) in-line replacements to reduce computation time.

Standards based implementations require bounds checking and accuracy specification.

Ground rules for spatial operations algorithm development

A Spatial Operation (SO) for Earth specific applications is a coordinate transformation, a coordinate conversion, an azimuth determination, a distance calculation or other computations associated with elliptical trigonometry and map projections.

GR 1: Consider the current computational environment:

- a surfeit of low cost Dynamic Random Access Memory (DRAM),
- high speed cache available,
- super scalar architectures for pipelined (parallel processing),
- very high speed processing of some standard mathematical functions,
- improved compilers consistent with pipelined processing,
- operating systems with dynamic optimization of resources,
- IEEE double precision as nominal processing mode.

GR 2: Consider the domain of application:

- develop for all regions but recognize that specific regions or applications may require different approaches,
- do not lose sight of the fact that majority of applications are in the near Earth region [-12, 35] kilometres (in geodetic height). This region encompasses the lowest point of Earth bathymetry (Marianas Trench) to the limit of air breather flight.

Ground rules for spatial operations algorithm development (cont'd)

GR 3: Define a meaningful error measure:

- use a Euclidean position error metric where this makes sense,
- convert angular errors to position errors,
- use a maximum position error over the *entire* region of interest.
- avoid the use of average absolute component errors.

GR 4: Avoid excessive accuracy:

- the SEDRIS goal is a maximum of 1mm position error over a region,
- avoid algorithms designed to yield errors like 10^{-08} or 10^{-20} if computational resources are required to achieve such accuracies.

GR 5: Set an acceptable error goal *before* starting development:

- for the SEDRIS accuracy goal, 0,000 9 mm, is < 0,022 860 in, is < 1/32 in,
- note that when the maximum position error is small the component errors are small,
- when angular errors are changed to position errors a small position error usually implies a very small angular error.

Ground rules for spatial operations algorithm development (cont'd)

GR 6: Use mathematical formulations consistent with GR1:

- use low order piecewise continuous representations when possible (popularly called table look up). This exploits low cost DRAM,
- do not use in-line table look up for square root or transcendental functions. It is virtually impossible to compete with built in system functions for these,
- do use global approximations or piecewise approximation if there is no high performance system routine. Note that, for example, sin(x) is not sin(f(x)). Depending on f(x), sin(f(x)) may be better handled by piecewise approximation,
- if forced to use a series, use only as many terms as required near the expansion point and more further away from the expansion point,
- for iterative methods determine the number of iterations required via error analysis and do not use expensive termination tests.

Ground rules for spatial operations algorithm development (cont'd)

GR 7: Use good programming practices:

- modern compilers can repair bad programming practices but do not depend on that,
- compute global constants once at start up,
- compute local constants once at the highest loop level,
- nest polynomials, etc,
- coding of the publication form of an algorithm is some times referred to as a *naïve* implementation. *Never* code the publication form of an algorithm without considering the computational aspects. Use trigonometric identities to simplify, avoid power functions, logarithms, exponentials by re-formulation. Seek competent counsel if you lack experience,
- naïve formulations and implementations are common in the literature and even in authoritative codes. Many times the conclusions of a publication are completely wrong due to the use of naïve implementations.

Ground rules for spatial operations algorithm development (cont'd)

GR 8: Design to context:

- algorithms for computing SOs should be designed for the context in which they are used. SO algorithms are usually not employed in isolation. Many SO computations are followed immediately by several more to form what is called a *chain*.
- all variables needed for the chain should be retained for subsequent use,
- designing for context may determine which of several procedures is best.

Example for an ERM: Some algorithms for converting geocentric coordinates to geodetic naturally compute the trigonometric functions of latitude and the radius of curvature in the prime vertical to full accuracy as part of the process. If this conversion is followed by a transformation to a map projection these variables will be needed. A good design for the first conversion will retain them. On the other hand some popular algorithms for the first conversion do not naturally produce these variables and to get them require more relatively expensive calculations. This should influence the choice of algorithm.

Ground rules for spatial operations algorithm development (cont'd)

GR 9: Software verification & computational error assessment:

- it is good practice to use authoritative data to confirm results, but do not be overly surprised that such data bases themselves are too sparse and are sometimes not accurate enough,
- all formulations in the SRM have closed form solutions in at least the forward direction. These can be used to *automatically* generate a dense set of test points,
- forming localized dense test sets near singular points is a good practice,
- one result of dense testing is to verify that the implementation is working properly over its intended area of application. Formulation and coding errors will be evident when dense test sets are used,
- converting a point, then converting that back to the original and then repeating the process a large number of times is called circular testing. Circular testing will most likely yield a divergent sequence of results. This is because round-off error, truncation error and other approximation errors are not symmetric. Circular testing is not a valid test procedure.

Ground rules for spatial operations algorithm development (cont'd)

GR 10: Performance testing can be difficult :

- in the literature, operation counts and counts of square root and transcendental function calls are often used to compare algorithms. Such a practice is generally a very poor predictor of performance and should only be used for rough judgments,
- sometimes algorithms are tested in isolation. A loop is coded and the loop time determined from system timing utilities. The candidate algorithm is entered into the loop, timed for a large number of replications, the loop time removed and the average processing time is computed. This kind of testing is called *in situ* testing. *In situ* testing can also yield misleading results, especially for the systems as outlined in GR 1. For the simple programs used *in situ* testing there is no competition for system resources and all the power of a super-scalar architecture will be used. The performance results may be very misleading.
- *In vivo* testing is performance testing in the context of a much larger application program. In this case there will be competition for system resources and algorithm performance may be impacted,
- while *in vivo* testing is always desirable it may not be possible early in a large development program. The developer must then use the other forms of assessment that are available while keeping in mind what might happen in the final product,
- it is always advisable to perform *in vivo* tests as a final confirmation of the algorithm performance.

Spatial Operations

Quality

Accuracy

Conformance Testing

Quality assurance for spatial operations

- Specifying the quality of spatial operations requires *error determination.*
- The meaning of "error" depends on the context and application domain.
- Potential sources of error in spatial operations include:
  - formulation error, approximation error, round-off error, truncation error and other errors associated with implementing the spatial operation.
- Errors of this nature should not be confused with errors arising from modeling the true shape of a celestial object (real or virtual) by an approximate shape.
- In the SRM, the reference surface used to approximate the shape of a celestial object, is assumed to be exact.
  - How well a reference surface approximates the shape of a celestial object is *outside the scope* of the SRM.

The meaning of accuracy is context dependent

- In geodesy a reference set of measured points is used to develop an Earth Reference Model (ERM)--a mathematical model of the Earth.
- ERMs provide a mathematical framework for spatial referencing.
- The amount of position error introduced by using an imperfect reference set of measured points is a major technical issue in geodesy but is outside the scope of the SRM.
- In the context of the SRM, reference models of spatial objects are taken to be exact when forming spatial reference frames.
- Operations on spatial reference frames may contain inaccuracies due to:
  - the use of truncated power series representations,
  - round off error,
  - approximations made to reduce computational complexity to speed up computations.

Accuracy Or Error In The Context Of the SRM Refers To Computational Error And Excludes Measurement Error Effects.

Conformance verification for the SEDRIS implementation

- A fundamental discriminator for level of conformance verification is the numerical difference between a result obtained from the mathematical specifications in the SRM and the corresponding result obtained by an implementation.
  - This requires that there be a *Reference Implementation.*
  - A data set produced by the Reference Implementation to support conformance testing is called a *Reference Data Test Set.*
  - For this purpose, the data points obtained from the RDS are taken to be the exact or true data points.
- Quantities in the Reference Data Test Set are called *test points*.
  - Each implementation to be tested shall have provisions for recording the information needed to compare to the Reference Data Test Set to support of conformance testing
  - This set of output data is called a *Test Data Set.*
  - The difference between the data points of an *Reference Data Test Set* and the *Test Data Set* of a particular implementation is referred to as an *error.*

Acceptance testing methodology

- It is desirable to have the number of data points be *relatively large* and *uniformly distributed* over the domain of the operation being evaluated
  - Else critical points, where implementation is flawed, may be missed.
  - The size and spatial distribution of values in the Reference Data Test Set is important and is operation dependent.
  - Once the Reference Data Test Set is specified it is relatively easy to evaluate the appropriate error metric over the whole set of values and to find the maximum error on the Reference Data Test Set.
  - This maximum error is used to determine the level of compliance of a particular implementation.

- Levels of acceptance
  - A particular implementation should not be required to meet the standard at the highest level if this induces unnecessary complexity and cost penalties.
  - In some applications, users may choose to simplify or approximate the formulations to reduce implementation and computational complexity and in particular to reduce computer processing time.
  - In doing so, they are willing to accept some degradation in accuracy for a particular application domain.

The Structure of SRM ISO Standard

ISO/IEC 18026

(Committee Draft)

Structure of the SRM ISO standard
(1 of 2)

The following clauses make up this International Standard:

- *Scope* defines the problem area that this International Standard addresses.
- *Normative references* lists the standards normatively referenced in this International Standard.
- *Terms, symbols and abbreviations* contains the glossary of terminology not other defined in other clauses, symbology and abbreviations used in this International Standard.
- *Concepts* contains introduction to the concepts used in this International Standard.
- *Coordinate systems* defines a common framework for the definition of abstract coordinate systems.
- *Temporal coordinate systems* specifies how to define coordinate systems for time.
- *Reference Datums, embeddings and Object Reference Models* specifies ways of measurably identifying origins and other basic locations in "space" or spatial objects.
- *Spatial Reference Frames* defines spatial coordinates systems and the Spatial Reference Frames included in this International Standard.
- *Vertical offset surfaces* specifies surfaces which model other aspects of a spatial object such as geiods.

Structure of the SRM ISO standard
(2 of 2)

- *Operations* specifies how to convert coordinates and other geometric concepts among the Spatial Reference Frames defined in this International Standard.
- *Application Program Interface* defines the functional interface for the operations included in this International Standard.
- *Registration* a defined process for expanding the standardized SRM.
- *Conformance* defines what it means for an implementation to conform to this International Standard.

And several annexes are including:

- *Annex: Mathematical Foundations* -contains required terminology and concepts.
- *Annex: Implementation notes* -contains some relevant computational algorithms.
- *Reference objects and parameters* lists the parameters defining reference objects used in this International Standard including reference ellipsoids and ORMs.
- *Annex: Change and deprecation plan* - defines a process for change and deprecation of both standardized and registered SRM concepts to ensure the long-term coherence and orderly evolution of concepts.
- *Annex: Conformance testing* -provides guidelines for developing conformance requirements.
- *Bibliography* lists the informative, non-standard documents referenced in this International Standard.
- *Terminology index* index to terms defined in the body of the text.

SRM concept management

- To support continuous evolution and growth of the SRM International Standard, a process of *registration* is defined, allowing for the timely addition of new concepts to the SRM.
  - **Registration shall not be used to modify any existing standardized or registered SRM concept.**
  - **New SRM concepts are registered using the established procedures of the ISO International Registration Authority for Graphical Items.**
  - **These procedures require the proposer to supply all information for a new SRM concept except for its code.**
  - **The code shall be assigned, and the code space managed, by the ISO International Registration Authority for Graphical Items.**
- The following types of SRM concepts may be registered:
  - **CSs,**
  - **RDs,**
  - **RDSs,**
  - **ORMs,**
  - **SRF Instances, and**
  - **SRF Sets.**
- Additionally, a process of *deprecation* will be defined for designating out-dated or inappropriate concepts from the SRM.

Further reading

1. "*Handbook for Transformation of Datums, Projections, Grids and Common Coordinate Systems*", U.S. Army Corps of Engineers, Topographic Engineering Center, TEC-SR-7, 1998.

2. "*Department of Defense World Geodetic System 1984*", National Imagery and Mapping Agency, Third Edition, TR8350.2, 1997.

3. "*Geodesy for the Layman*", National Imagery and Mapping Agency, on-line at http://164.214.59/geospatial/products/GandG/geolay/toc.htm.

4. Richard H. Rapp, "*Geometric Geodesy Part I & II*", The Ohio State University, Dept. of Geodetic Science & Surveying, 1993.

5. John P. Snyder, "*Map Projections -- A Working Manual*", U.S. Geological Survey Professional Paper 1395, 1987.

6. Paul D. Thomas, "*Conformal Projections in Geodesy and Cartography*", U.S. Department of Commerce, Coast and Geodetic Survey, Special Publication 251.