Numerical Weather Prediction in 75 Minutes

Learning Outcomes

Following this lecture, students will be able to:

- Provide a basic description of a modern numerical weather prediction model's components.
- Recognize that many aspects of numerical weather prediction models are approximations, each of which contribute to forecast errors.

Introductory Remarks

How can "numerical weather prediction" be defined? Let us consider the dictionary definitions for each term in the phrase, numerical weather prediction:

- **Numerical**: using data or numbers; in reference to the methods used (i.e., the solving of mathematical equations).
- **Weather**: the state of the atmosphere with respect to wind, temperature, moisture, pressure, etc.; also referring to the time scale (e.g., in contrast to climate).
- **Prediction**: an act of declaration of a future event; more generally, a forecast of a past, present, or future event.

Thus, we define numerical weather prediction as the use of numerical methods to solve equations governing atmospheric motions and processes to obtain a forecast on weather spatiotemporal scales. Numerical weather prediction can be further broken down into two classes:

- **Dynamical**: a class of models in which the equations being solved are rooted in the primitive equations (e.g., the momentum equations, thermodynamic equation, ideal gas law, mass continuity, water vapor).
- **Statistical**: a class of models in which the equations being solved are based on statistical relationships between atmospheric variables at one time and the forecast at a subsequent time, including those such as Pangu-Weather that are derived from artificial intelligence methods such as deep learning.

The delineation between the two is not always clear. For instance, fine-scale processes in dynamical models are typically represented by *parameterizations*, many of which are empirically derived and thus rooted in statistical relationships. Further, statistical methods can be applied to dynamical-model outputs to refine or otherwise improve upon their forecasts. Conversely, some statistical models incorporate the primitive equations into their training procedures (e.g., as forecast constraints).

This course focuses on dynamical numerical weather prediction models. Although the European Center for Medium-range Weather Forecasts, other operational weather modeling centers, and private-sector entities are actively testing artificial-intelligence–based models within their forecasting workflows, this remains a nascent, rapidly evolving area of the field.

Numerical (or NWP) models are developed based upon accepted *theories* and rely upon *observations* to provide both initial and verification data so to conduct and verify a forecast. But, how do we solve the equations that govern atmospheric motions and processes? How do we provide the model with observations to initialize (or start) the model forecast? These questions seem as if they should have simple answers, but they unfortunately do not. There are multiple ways of answering each question, each of which has its own strengths and weaknesses. Indeed, the more one comes to find out just how many approximations or simplifications comprise a numerical model, the more one may become surprised at just how well they can perform!

Therein lies a key point – how well they **can** perform? There is an adage that states that you can obtain **any** forecast that you desire using a numerical model, and to some extent that is true. It is easy to run a model and obtain a forecast, even on a desktop or laptop computer, given current hardware capabilities and the recent advent of containerized models. Data to initialize these forecasts are also abundant, and many of them are free. It is far more difficult, however, to obtain a reasonably accurate forecast. To do so requires a deep understanding of how the model works and how it is best configured for the desired application, with details as small as a single value in a text file often having a large impact on the subsequent forecast!

Of course, this is all complicated by two factors:

- Despite several decades' worth of advances, we still do not know – and probably never will know – what is *the* best model configuration. It is not clear that one does or should exist.
- Even if we could identify a 'best' configuration, all models provide imperfect forecasts that, even if they verify well, are likely 'right for the wrong reasons.' In other words, *the ability to predict does not necessarily imply predictability*!

In this class, it is my goal to introduce how dynamical models are constructed; the numerical methods used to obtain forecasts and some of the shortcomings inherent to those methods; why and how certain physical processes are parameterized; how numerical models are initialized; and teach you to appropriately configure, run, and analyze the output from a numerical model.

To first order, the primitive equations form the underlying equation set for dynamical NWP. It is not as simple as solving the primitive equations in the form that we are most familiar with to obtain a forecast, however. We will begin to directly illustrate this in an upcoming lecture. For now, however, we wish to introduce some general considerations underlying NWP. Much of the remainder of the semester will be spent examining both the fundamental and practical implications of these considerations.

Discretizing the Fluid Atmosphere

Let us consider the atmosphere and its properties to be a fluid. This fluid, as we know, is comprised of many molecules. How many?

- **Total atmospheric mass**: approximately 5.148 x 10^{21} g, per the study of Trenberth and Smith (2005, *J. Climate*).
- **Mass of 1 mol of dry air**: 28.97 g
  - **Total mol of air**: 5.148 x 10^{21} g divided by 28.97 g mol^{-1} = 1.78 x 10^{20} mol.
- **Number of molecules in a mol**: Avagadro's number, or 6.022 x 10^{23} mol^{-1}.
  - **Total molecules of air**: 1.78 x 10^{20} mol multiplied by 6.022 x 10^{23} mol^{-1} = 1.07 x 10^{44} molecules.

Even if there were to exist global observations of pressure, temperature, wind, and other relevant atmospheric fields at the molecular level, we simply cannot solve for this many elements in a reasonable amount of time.

Rather, the atmosphere must be **discretized**, or divided into a smaller number of finite elements. This discretization occurs in all three spatial dimensions and, as we will see, time. The two most popular means of spatial discretization are **grid-based** and **spectral** methods:

- **Grid-based methods** discretize the atmosphere into a grid. Models within this grouping may solve the equations at discrete grid points (grid-point models) or over the volumes carved out by how the grid points are arranged (finite-volume models). Grid volumes are most commonly in the form of cubes, although some newer models use hexagonal or icosahedral prisms to represent these volumes. NOAA's Global Forecast System and the UK Met Office's Unified Model are two examples of grid-based models.
- **Spectral methods** discretize the atmosphere into the superposition of a finite number of waves (*O*(500-1500)) of varying wavelength. The ECMWF's operational models are examples of spectral-method–based models.

Small grid volumes and/or a larger number of waves provide for higher resolution (or finer discretization) at the cost of greater computational expense. We focus primarily on grid-based methods in this course, although all concepts to be covered apart from how the numerical equations are solved in spatial dimensions apply to spectral-based models as well.

Let us consider an example of a grid-point model, the UK Met Office's Unified Model. This model is run four times daily over a global domain. It uses 2,560 grid points in the east-west direction and 1,920 grid points in the north-south direction, corresponding to an approximate horizontal grid spacing of 10 km in the midlatitudes. The model also uses 70 non-evenly spaced vertical levels between the surface and approximately 80 km above the surface. As a result, the Unified Model discretizes the atmosphere into 2560 x 1920 x 70, or 3.44 x 10^8 grid points. Indeed, this is a lot of grid points, requiring a supercomputer valued in the tens of millions of dollars on which to run and obtain a forecast in a reasonably short amount of time. However, it is still 36 orders of magnitude smaller than the total number of air molecules in the atmosphere.

Model Resolution and Parameterizations

Whether using a grid-point, spectral, or type of model discretization, the atmosphere is discretized to ensure that the smallest meteorological feature of interest (for the desired application, and considering available computational resources and time constraints) is adequately **resolved**.

What does it mean to resolve a feature? Let's consider a grid-point discretization. Think about shortwaves and longwaves in the geopotential height field, the former having wavelengths of ~500-1000 km and the latter having wavelengths of up to 10000 km. Let us discretize the atmosphere such that there is one grid point every 100 km. For the shortwave, this means that there are six grid points from east to west across the wave (Fig. 1), which is barely enough to crudely identify a trough and ridge. For the longwave, however, there are roughly ten times as many grid points across the wave. Thus, the structure of the wave will be better represented, or resolved, for the longwave than for the shortwave.

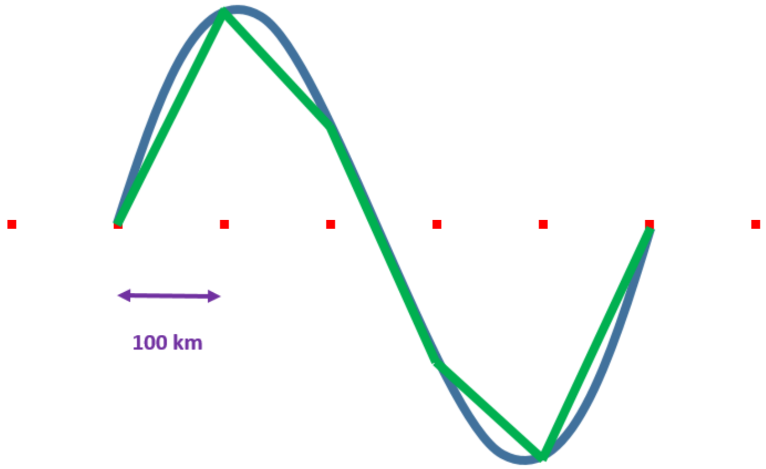

Figure 1. The amplitude of a shortwave trough of wavelength λ = 500 km (blue) is represented as it would be discretized on a one-dimensional horizontal grid (red) with horizontal grid spacing Δx = 100 km (green).

For NWP, we solve the primitive equations formally only for scales that the model can resolve. But, though we cannot resolve them, the smaller unresolved scales are still important. Indeed, it is on those scales that processes such as turbulent vertical mixing in the planetary boundary layer, water substance phase changes within clouds, and (in many cases) deep, moist convection occur. Further, the unresolved scales evolve in and subsequently influence the resolved scales.

We account for what occurs on the unresolved scales with **parameterizations**. A parameterization empirically approximates unresolved scales as a function of known or hypothesized relationships with resolved scales. We will consider parameterizations again briefly later in this lecture and in more detail mid-semester.

How do we determine what a model can resolve, or its resolution? In spectral models, resolved features have a wavelength longer than the shortest wave retained by the spectral discretization. It is a little less clear for grid-point models. As Fig. 1 shows, while we can represent a wave-like feature with just a few grid points, it may not be well resolved. To help address this issue, we can distinguish **resolution** and **grid spacing**:

- **Grid spacing (Δx)**: the distance between two grid elements or points.
- **Resolution**: the scale of the smallest wave/feature that can be well-represented on the model grid.

The grid spacing is smaller than the resolution, as more than two points are needed to represent a feature on a grid. We can determine a model's resolution by assessing how well its kinetic-energy spectrum as a function of horizontal scale compares to that derived from theory. As we will see later this semester, the numerical methods used in dynamical NWP models typically result in the modeled kinetic energy dropping off quicker than expected from theory at scales between the model's resolution and grid spacing (Fig. 2). The horizontal scale at which this drop-off begins defines its resolution, or sometimes referred to as its **effective resolution**.

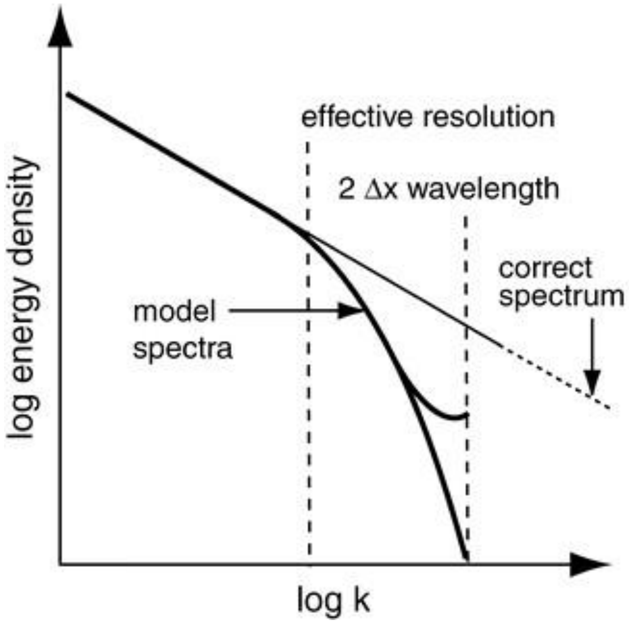

Figure 2. Schematic depicting the kinetic energy spectra for two sets of model simulations: those conducted at horizontal grid spacings in the mesoscale (thick black line with upward-curved tail) and those conducted at horizontal grid spacings on the micro- or synoptic-scale (thick black line that decays exponentially). The *x*-axis depicts the natural logarithm of the wavenumber *k,* such that higher values of log *k* denote smaller spatial scales. The *y*-axis depicts the natural logarithm of the spectral energy density; higher values denote greater kinetic energy. In these examples, the model spectra match the correct spectrum up to the effective resolution but are thereafter damped. Figure obtained from Skamarock (2004, *Mon. Wea. Rev.*), their Fig. 10.

For modern grid-point models such as the grid-point-based WRF-ARW (Skamarock 2004, *Mon. Wea. Rev.*) and the unstructured-grid-based finite-volume MPAS (Skamarock et al. 2014, *J. Atmos. Sci.*) models, their effective resolution is approximately 6-7Δx. Thus, it takes seven to eight grid points to effectively resolve an atmospheric feature in these models. Applied to the example in Fig. 1, the shortwave would not be well-resolved by the WRF-ARW or MPAS models, but the longwave would be well-resolved by these models.

Note that these concepts are only related to how well a feature can be represented by a model. They are not related to the observations used to initialize a model forecast. These observations are typically irregularly spaced and require interpolation to the model grid (and model variables) when they are used to help initialize a model forecast. How this is done is briefly discussed later in this lecture and in more detail at the end of the semester.

Discretization can be non-uniform in both the horizontal and the vertical directions. Variable horizontal discretization, such as that depicted in Fig. 3, is typically used to provide higher resolution over a region of interest, often to avoid the computational expense associated with running the model at high resolution over a large area extending beyond your region of interest. Certain grid discretizations, such as the latitude-longitude grid, inherently provide variable resolution, with coarser resolution (1° longitude = 111 km) near the Equator and finer resolution (1° longitude < 111 km) at higher latitudes, but also present numerical challenges near the poles.

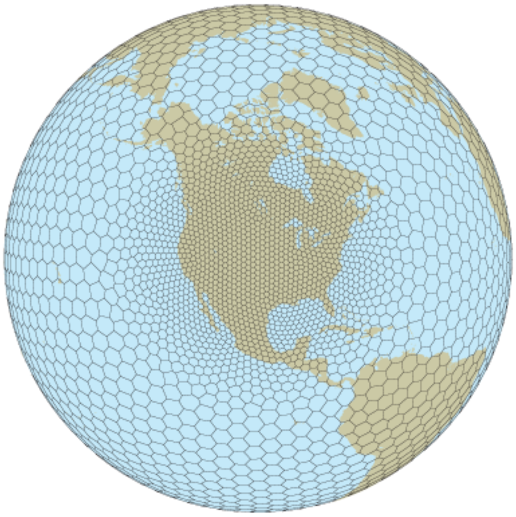

Figure 3. An example of a variable resolution model grid discretization, with higher resolution across North America and lower resolution elsewhere, as employed in the Model for Prediction Across Scales, or MPAS. Image obtained from https://mpas-dev.github.io/.

Most modern models use variable vertical discretization, with finer discretization near the surface (and sometimes near the tropopause) and coarser discretization in the midtroposphere and stratosphere. This is motivated by the sharp vertical gradients in meteorological quantities typically near the surface and at the tropopause. Most modern models implement variable vertical discretization using a hybrid terrain-following vertical coordinate. Near the surface, the associated vertical levels parallel the local terrain height. In the mid to upper troposphere, the associated vertical levels gradually transition to follow constant height, isobaric, or isentropic surfaces depending on the model. An example is given in Fig. 4. Because models do not use a vertical coordinate that appears in the standard forms of the primitive equations, these equations must be transformed so that they can be applied using the model's chosen vertical coordinate.

Modern NWP models use anywhere between 30 and 100 vertical levels between the surface and the chosen model top (typically in the stratosphere or higher, placed above the uppermost level of meteorological interest). Models that have finer horizontal discretization typically also have finer vertical discretization. As in the horizontal, it takes more than two vertical levels to truly resolve a feature in the vertical, such as an inversion, front, or jet.

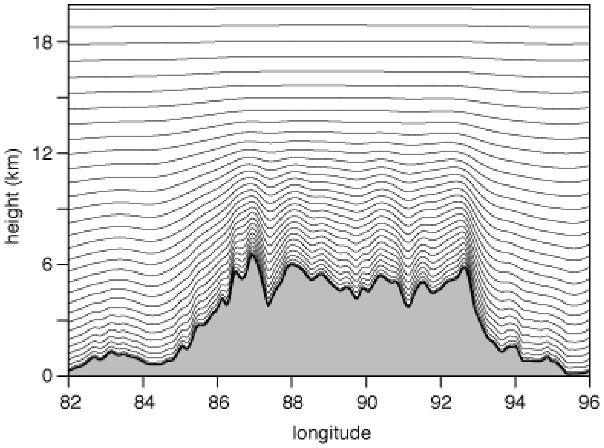

Figure 4. An example of a hybrid terrain-following vertical coordinate over a region of sloped topography. In this example, representative of the vertical coordinate employed in the Model for Prediction Across Scales, vertical levels follow the terrain below approximately 12 km above sea level. At higher altitudes, vertical levels closely resemble constant height surfaces. Image obtained from https://mpas-dev.github.io/.

Numerical Methods

Once we have discretized the atmosphere, whether using a grid, the spectral method, or some other means, how do we solve the primitive equations for that discretization? For spectral models, Fourier series and Fourier-Legendre transforms are used to transform horizontal variability (partial derivatives) between physical and spectral/wave space.

For grid-based models, finite-difference approximations to partial derivatives are used to solve the primitive equations. Mathematically speaking, the partial derivative of some generic field *f* with respect to some generic variable *x* can be expressed as:

$$\frac{\partial f}{\partial x} = \lim_{\Delta x \to 0} \frac{\Delta f}{\Delta x} \qquad (1)$$

In other words, $\frac{\partial f}{\partial x}$ is equal to the value of $\frac{\Delta f}{\Delta x}$ as Δx approaches (but does not equal) zero.

Thus, for small (or finite) values of Δx, we can approximate $\frac{\partial f}{\partial x}$ by $\frac{\Delta f}{\Delta x}$.

Taylor series expansions of a function are used to obtain finite difference approximations. The Taylor series expansion of *f(x)* about *x* = *b*, where *b* is some generic point, is given by:

$$f(x) = f(b) + f'(b)(x-b) + \frac{f''(b)}{2!}(x-b)^2 + \frac{f'''(b)}{3!}(x-b)^3 + \ldots \qquad (2)$$

In other words, *f(x)* is equal to the value of *f(x)* at *x* = *b* plus a series of higher-order terms, each of which has a different derivative (primes), exponent on *x* - *b*, and factorial (!) order.

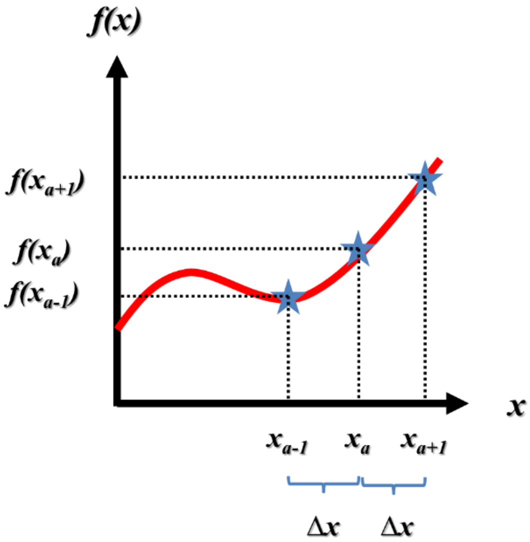

Figure 5. Graphical depiction of a generic function *f(x)* evaluated at three points.

Let us consider the case where $x = x_{a+1}$ and $b = x_a$, as in Fig. 5. The distance $x - b$, or $x_{a+1} - x_a$, is equal to Δx. Conversely, let us consider the case where $x = x_{a-1}$ and $b = x_a$. The distance $x - b$, or $x_{a+1} - x_a$, is equal to $-\Delta x$. Using this information, we can expand (2) for each of these two cases:

$$f(x_{a+1}) = f(x_a) + f'(x_a)\Delta x + \frac{f''(x_a)}{2!}(\Delta x)^2 + \frac{f'''(x_a)}{3!}(\Delta x)^3 + \ldots \qquad (3)$$

$$f(x_{a-1}) = f(x_a) - f'(x_a)\Delta x + \frac{f''(x_a)}{2!}(\Delta x)^2 - \frac{f'''(x_a)}{3!}(\Delta x)^3 + \ldots \qquad (4)$$

Note the similar appearance of (3) and (4) apart from the leading negative signs on the first and third order terms in (4). These arise because $x - b = -\Delta x$ here, as noted above.

From (3) and (4), we are interested in the value of $f'(x_a)$. This is equivalent to $\frac{\partial f}{\partial x}$. We can use (3) and (4) to obtain an expression for this term; we simply need to subtract (4) from (3). Doing so, we obtain the following:

$$f(x_{a+1}) - f(x_{a-1}) = 2f'(x_a)\Delta x + \frac{2f'''(x_a)}{3!}(\Delta x)^3 + \ldots\text{(odd order terms)} \ldots \qquad (5)$$

Note how the terms on the right-hand side that lack Δx and the terms involving $(\Delta x)^2$ in (3) and (4) cancel out in this operation. If we rearrange (5) and solve for *f'(x$_a$)*, we obtain:

$$f'(x_a) = \frac{f(x_{a+1}) - f(x_{a-1})}{2\Delta x} - \frac{f'''(x_a)}{3!}(\Delta x)^2 + \ldots \qquad (6)$$

At this point, we wish to neglect all terms higher than the first-order term from (6). Doing so, we are left with:

$$f'(x_a) = \frac{f(x_{a+1}) - f(x_{a-1})}{2\Delta x} \qquad (7)$$

Equation (7) is what is known as a *centered finite difference*. This expression is said to be *second-order accurate* because terms involving $(\Delta x)^2$ and higher-order powers are dropped, or **truncated**, as being small relative to the retained lower-order terms.

We are likely also already familiar with first-order–accurate forward (8) and backward (9) finite differences. The forward difference is obtained by solving (3) for $f'(x_a)$, whereas the backward difference is obtained by solving (4) for $f'(x_a)$:

$$f'(x_a) = \frac{f(x_{a+1}) - f(x_a)}{\Delta x} \qquad (8)$$

$$f'(x_a) = \frac{f(x_a) - f(x_{a-1})}{\Delta x} \qquad (9)$$

Higher-order (i.e., more accurate) finite-difference expressions can be obtained similarly. These involve using even more points to calculate the finite-difference approximation to the first partial derivative. Modern NWP models typically use fourth-, fifth-, or even sixth-order–accurate finite difference approximations in space.

In the context of finite difference methods, however, each approximation has a unique impact upon the forecast. The most straightforward of these to conceptualize is **truncation error**, such as that depicted in Fig. 6. Dropping higher-order terms from the expanded Taylor series, though they are comparatively small, reduces the approximation's accuracy. As NWP model forecasts can be quite sensitive to very small differences in initial or forecast value, this can impact forecast quality. For grid-based models, truncation error is particularly problematic for features that can be represented but not resolved by the model, or those lying between the grid spacing and effective resolution. For spectral models, truncation error manifests for features with wavelengths smaller than that of the smallest wave represented by the model.

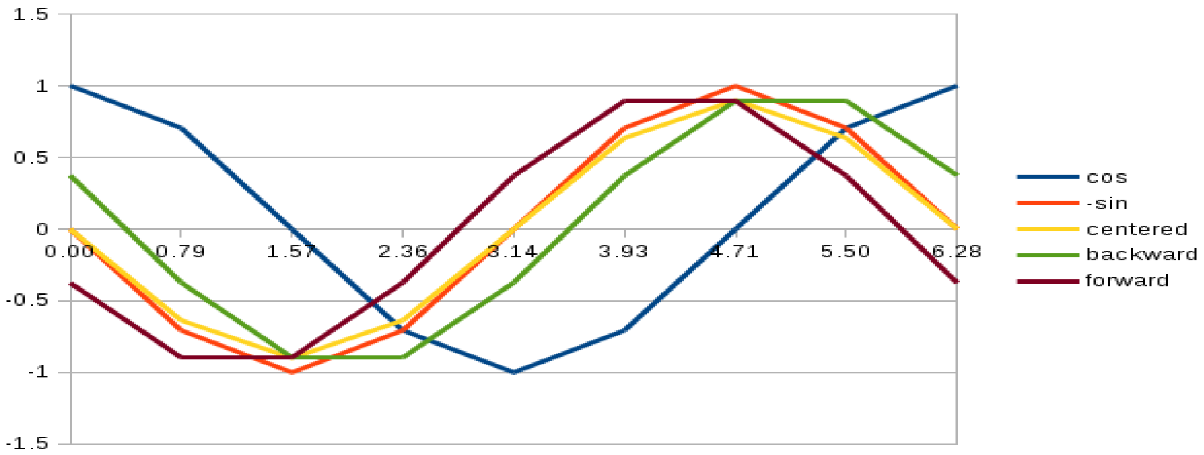

Figure 6. An example of truncation error for the function *f*(x) = cos(x) over a grid with nine grid points between 0 and 2π. The function *f*(x) = cos(x) is plotted in blue. Its first partial derivative, or –sin(x), is plotted in red; this is the **exact** solution for *f*'(x). Three finite difference approximations for the first partial derivative, the centered, forward, and backward approximations, are plotted in yellow, brown, and green, respectively. None provide exact solutions, with the centered approximation (the highest-order–accurate of the three) best resembling the exact solution. The extent to which each approximation differs from the exact solution can be thought of as a metric of the truncation error.

What about temporal discretization? Whether for grid-point or spectral models, finite differences are also used to advance from one time step to another. There exist multiple methods for temporal finite differencing, each with its own accuracy and numerical considerations. Let us start by considering a centered scheme in time, known as the "leapfrog" scheme; we will discuss other, more accurate methods later in the semester.

Here, the partial derivative of some function *f* with respect to time, at time *t*, is expressed as:

$$\frac{\partial f}{\partial t} = \frac{f_{t+1} - f_{t-1}}{2\Delta t} \tag{10}$$

where this term is then equal to the sum of all forcing terms (at time *t*) that act to influence the local value of *f*. By solving that equation for f_{t+1}, the value of *f* at the next time can be obtained: here, equal to f_{t-1} plus 2Δt times the sum of the forcing terms.

How does this work in practice? Consider the schematic in Fig. 7. At *t* = 0, a forward difference is used to go from *t* = 0 to *t* = 1. The value of *f* at *t* = 1 is equal to that at *t* = 0 plus the forcing upon *f* (at *t* = 0) multiplied by the timestep Δ*t*. Then, the leapfrog scheme (10) is used to advance to *t* = 2. The value of *f* at *t* = 2 is equal to that at *t* = 0 plus the forcing upon *f* (at *t* = 1) multiplied by twice the timestep 2Δt. The value of f at t = 3 is then equal to that at t = 1 plus the forcing upon f (at t = 2) multiplied by twice the timestep 2Δt, and so on.

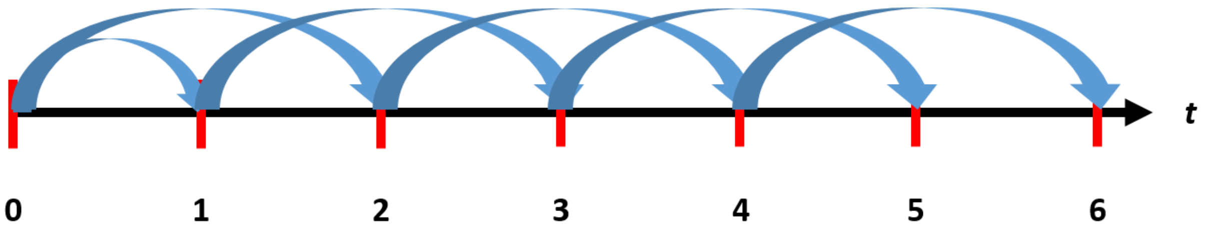

Figure 7. Graphical representation of the leapfrog temporal differencing scheme. Please refer to the text above for details.

Since the model equations are not solved at every possible time (e.g., every infinitesimal difference in time), temporal discretization is also a source of truncation error. Atmospheric processes that evolve on shorter timescales than those of the model time step cannot directly be represented by the model equations.

Numerical Stability: The Courant Number and CFL Condition

How is a model's temporal discretization determined? Whereas we can typically specify the horizontal grid spacing as a function of what we want to be able to resolve within our forecasts, we have less control over the model's time step Δt. This is because of the concept of **numerical stability**, the specifics of which depend on the chosen finite-differencing schemes.

For spatial and temporal differencing schemes that produce numerically stable solutions – ones that evolve physically rather than numerically grow exponentially during the model forecast – there exists a limiting value of the **Courant number**. In its most general of forms, the Courant number is expressed as:

$$C = \frac{U\Delta t}{\Delta x} \tag{11}$$

The Courant number is non-dimensional, where U is the translation speed of the fastest feature or wave on the model grid, Δx is the horizontal grid spacing, and Δt is the model time step. Each combination of spatial and temporal finite-differencing schemes has a different limiting factor of the Courant number known as the **CFL condition**, representing the maximum value of the Courant number that permits numerically stable model solutions. A general form of the CFL condition is $C \leq 1$. Conceptually, this implies that an air parcel cannot travel more than one grid distance in one model time step. Since U is governed by the prevailing meteorology and Δx is determined by the horizontal scales of the features that one desires to resolve on the model grid, Δt must be chosen to ensure that the CFL condition is met.

Let us consider the example of a jet streak with U = 100 m s^{-1}. On a model grid with horizontal grid spacing of 4 km, for $C \leq 1$, $\Delta t \leq 40$ s. In other words, we must solve the primitive equations across the model grid once every 40 s or less to maintain numerical stability. For a 24-h forecast, this involves 86400 s / 40 s = 2160 separate times that the primitive equations are solved. Over a continental United States domain, with 1150 x 750 x 50 grid points for Δx = 4 km,

- **Number of model grid points**: 1150 x 750 x 50 = 43,125,000
- **Number of model time steps**: 2,160
  - **Total calculations for one variable**: 43,125,000 x 2,160 = 9.315 x 10^{10}

Contrast this to a model configuration with 12 km horizontal grid spacing. For $C \leq 1$, $\Delta t \leq 120$ s, requiring that the primitive equations be solved only 720 times for a one-day forecast. Over a domain with the same continental United States extent as above,

- **Number of model grid points**: 383 x 250 x 50 = 4,787,500
- **Number of model time steps**: 720
  - **Total calculations for one variable**: 4,757,500 x 720 = 3.447 x 10^9

Thus, a longer time step can be used at coarser horizontal grid spacings to maintain numerical (or computational) stability. Of note, however, is that coarsening the horizontal grid spacing by a factor of 3 results in 27 times fewer calculations needed! There are 3 times fewer horizontal grid points in both the north-south and east-west directions and 3 times time steps needed, with 3*3*3 = 27 times fewer calculations.

Lateral Boundaries

A numerical model represents the atmosphere in three spatial dimensions. If the model is a **global model**, then the horizontal dimensions wrap back upon themselves. This is one instance of what are known as *periodic boundary conditions*, where what goes out one side comes back in on the other side. Conversely, models that simulate only part of the Earth are known as **limited-area models**. The east-west and north-south extents of such models are limited; thus, for such models, we need to know what is coming into and going out of the domain.

This is accomplished using *specified boundary conditions* on the lateral boundaries (where lateral refers to sides). These are generally drawn from the forecast and/or analysis of a global or larger-area model – possibly even one you run yourself! Such lateral boundary conditions are typically coarser than the horizontal grid spacing of the limited-area model and, as they are drawn from a different model, typically do not exactly agree with the forecast from the limited-area model. Thus, a transition and damping buffer zone between the lateral boundaries and the interior of the limited-area model domain is generally applied, such as that depicted in Fig. 8.

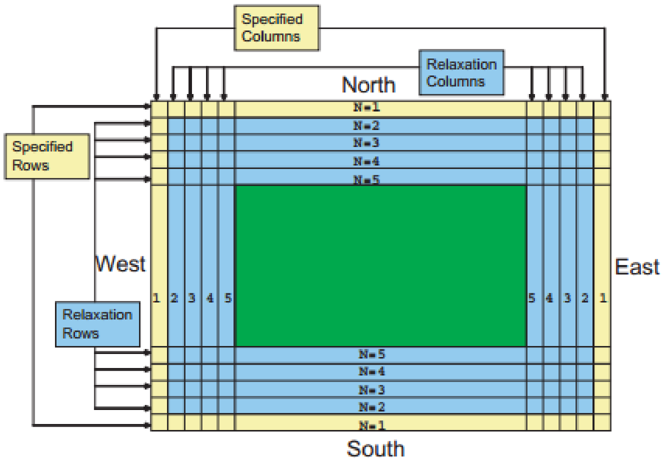

Figure 8. Representation of a five-point transition or buffer zone (blue) between the lateral boundaries of the limited-area model domain (yellow; generally interpolated from coarser model data to the spacing of the finer limited-area model domain) and the interior of the limited-area domain (green). Image obtained from https://opensky.ucar.edu/islandora/object/opensky:2898.

Because of using a transition zone near the limited-area model domain's lateral boundaries, the lateral boundaries are typically placed far away from the feature(s) and/or region(s) of most interest within the limited-area model domain.

Upper and Lower Boundaries

Both limited-area and global models have upper and lower boundaries. The upper boundary is typically specified well above the level of meteorological importance for your region and/or case. Most models incorporate an artificial damping layer in the uppermost few kilometers of the model domain to dampen vertically propagating waves so that they and their energy are not reflected downward when they impinge upon the top of the domain.

The bottom boundary is more challenging and, given that it is where we all live, more important to accurately represent on meteorological time scales. At a bare minimum, accurate topographic and land-use information must be used by a model. In addition, heat and moisture transport between the surface and model atmosphere must be accurately handled, as must the effects of surface friction. These are typically handled by **land-surface models** that are coupled to (i.e., run with two-way feedback with) the NWP model, with surface-layer formulations to represent the effects of what occurs in the first centimeters above the surface. Some models are also coupled to ocean models, which can represent how atmospheric forcings modify the oceanic state (and thus influence the atmospheric forecast). This is most important when air-sea interaction is important, such as for tropical cyclones.

Initial Conditions

A model cannot run if it does not know the values of its variables (wind, temperature, moisture, etc.) at its initial time, known as its **initial conditions**. In general, initial conditions should:

- Be as accurate (close to observations, acknowledging that observations themselves have uncertainty and error) as possible. Despite the non-linearity of the atmosphere, forecast quality is directly proportional to initialization quality.
- Represent features at and coarser than the model's effective resolution. If they only represent coarser features, the so-called **cold start**, then finer-scale features must "spin up" over the first 6-12 h of the model forecast. Those initial conditions that do represent such finer-scale features are known as **warm start** (partially resolved) or **hot start** (fully resolved) initial conditions.
- Adhere to an appropriate balance condition. For vertical motions, especially for coarser models, this is typically hydrostatic balance, even if the model itself is non-hydrostatic. For horizontal motions, this could be geostrophic, gradient, or some more complex non-linear form of balance. Note that not all models will balance or insist on the existence of balanced initial conditions. However, it is a good idea that they be at least close to balanced to mitigate the generation of artificial inertia-gravity waves (that attempt to establish or restore balance) early in the model forecast.

While it is theoretically possible to apply some form of objective-analysis routine to observations to obtain initial conditions, this is typically not done. Instead, **data assimilation** is used, wherein some "first guess" of the initial conditions drawn from an earlier model forecast is corrected via some means of assimilating available observations. Obtaining initial conditions in such a way is an example of **cycling**, as depicted in Fig. 9.

Most limited-area models use initial conditions generated from another model's initialization, such that their users do not need to perform data assimilation themselves. It is possible, however, for users to initialize forecasts with their own cycled model + data assimilation system. Coupling a model like WRF-ARW to a data assimilation package such as the Data Assimilation Research Testbed (DART) toolkit is one common manifestation of this.

There exist multiple methods underpinning modern data-assimilation methods, each of which are founded on the principle of minimizing error in the initial conditions relative to observations. As you might expect, each of these methods have their own strengths and weaknesses, which we will discuss in more detail at the end of the semester.

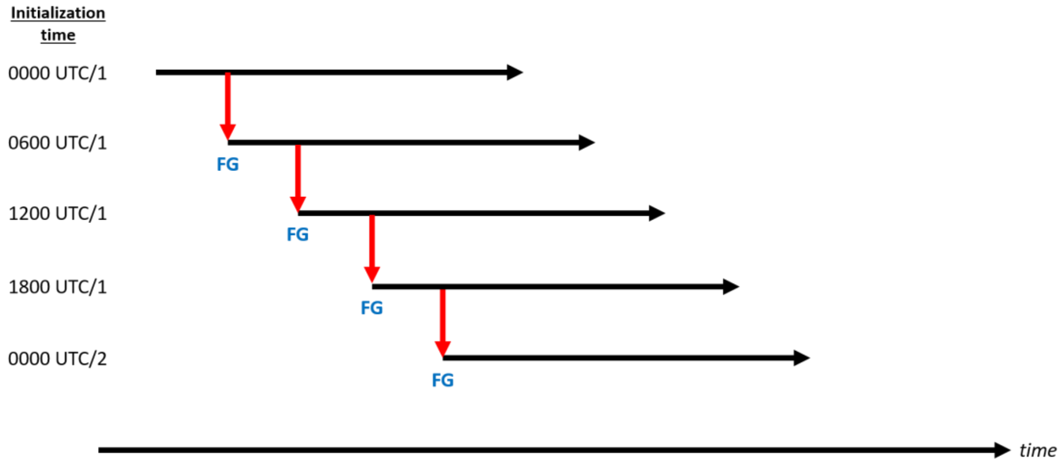

Figure 9. A schematic of forecast cycling, where in this case the 6-h forecast from a previous model run is used as the “first guess” (FG) for the next model run. Observations at the initialization time are assimilated to update and correct the “first guess” before the forecast is launched. This schematic represents a manifestation of three-dimensional data assimilation as temporal variability of the assimilated observations for a given model cycle is not considered (i.e., they are all assumed to occur at a single time).

Model Parameterizations

As noted earlier, every model has features that it cannot resolve, even though what happens on the unresolved scales may be quite important both societally and to model forecast quality. Such processes must be **parameterized** by the numerical model. There are three reasons to use a parameterization:

- **Scale**: The process or set of processes cannot be resolved on the model grid.
- **Complexity**: We can resolve the process or set of processes on the model grid but doing so would be computationally expensive.
- **Knowledge**: We do not know enough about the relevant process or set of processes to have reliable physical or dynamical equations that we can solve to resolve the process or set of processes.

Of these, the first is by far the primary reason for parameterization in modern models. While the last is also relevant, it generally is encapsulated within and superseded by the first. Not many processes fall within the second – complexity – and, even if they did, the improved forecast quality that usually results from explicitly treating these physical processes typically far outweighs the added computational cost.

Parameterization involves representing a process in terms of its known relationships to variables resolved on the model grid. An example of this is turbulence, which is a function of horizontal and vertical wind shear (mechanical turbulence) and buoyancy/stability (buoyant turbulence). Given model-resolved wind shear and stability, a parameterization can infer the properties of and resolved-scale effects of unresolved sub-grid-scale turbulence. Another example is clouds: given a profile of model-resolved moisture and stability, a parameterization can infer the properties of and resolved-scale effects of sub-grid-scale clouds, including determining if they are present.

The most common processes that are parameterized by modern NWP models include vertical turbulence (both in the planetary boundary layer and above), surface layer, shallow and deep cumulus convection (the latter particularly for coarser models), cloud microphysics (e.g., phase changes and precipitation growth), and radiation. Land-surface and ocean models also incorporate parameterizations of their own. There are also other types of parameterizations, such as stochastic and orographic-drag parameterizations. We will cover the physics and parameterization methods for physical parameterizations in detail later in the semester.

Parameterizations have traditionally been empirical formulations grounded in the known physical and dynamical characteristics of the parameterized process(es). Specifically, the equation structure is generally determined by the parameterization developers from physical understanding, whereas the tunable parameters in these equations are determined empirically using limited observations or output from models that explicitly resolve the parameterized processes. However, recent research suggests that artificial-intelligence techniques can identify better equation structures if given sufficient data from which to learn. That said, all parameterizations are approximations and thus are leading sources of model error.

There are many ways that each parameterized process or set of processes can be parameterized, each with its own assumptions, strengths, weaknesses, and complexity. For example, WRF-ARW v4.5.1 includes thirteen turbulence or planetary boundary layer, thirty cloud microphysics, fourteen cumulus convection, eight shortwave radiation, and eight longwave radiation parameterizations! More details on these are available at https://www2.mmm.ucar.edu/wrf/users/wrf_users_guide/build/html/physics.html.