LIUC eBook

Riprogettare il curriculum di matematica

Corso di Formazione Docenti Scuola Secondaria di Secondo Grado

Lorella Carimali
Luca Mari

LIUC Università Cattaneo

LIUC eBook, 1

Riprogettare il curriculum di matematica
Corso di Formazione Docenti Scuola Secondaria di Secondo Grado

Lorella Carimali, Luca Mari

LIUC Università Cattaneo
Castellanza 2013

Riprogettare il curriculum di matematica
Lorella Carimali, Luca Mari

Data di pubblicazione: Giugno 2013 - ISBN 978-88-908806-0-5

Indice

RIPROGETTARE IL CURRICULUM DI MATEMATICA
Corso di Formazione Docenti Scuola Secondaria di Secondo Grado

Lorella Carimali, Luca Mari

Premessa: per una didattica delle competenze

Una competenza (generale, di studio, di lavoro) si sviluppa in un contesto nel quale la persona è coinvolta, individualmente o collettivamente, nell'affrontare situazioni, nel portare a termine compiti, nel realizzare prodotti, nel risolvere problemi, che implicano l'attivazione e il coordinamento operativo di quanto sa, sa fare, sa essere o sa collaborare con altri.

La progettazione di un'attività formativa diretta allo sviluppo di competenze deve dunque tener conto della necessità che le conoscenze fondamentali da questa implicate siano acquisite in maniera significativa, cioè comprese e padroneggiate in modo adeguato, che le abilità richieste siano disponibili a un livello confacente di correttezza e di consapevolezza di quando e come utilizzarle, che si sostenga il desiderio di sviluppare conoscenze e abilità nell'affrontare compiti e attività che ne esigono l'attivazione e l'integrazione.

Ciò implica l'uso di metodi che introducano i nuclei fondamentali delle conoscenze e delle abilità e il loro progressivo padroneggiamento e che coinvolgano l'attività degli studenti nell'affrontare questioni e problemi di natura applicativa (relativi alla propria vita, alle altre discipline, alla vita sociale, …). La chiave di volta metodologica risulta essere un ambiente di lavoro nel quale si realizzano individualmente o collettivamente prodotti che richiedono un utilizzo intelligente di quanto studiato o sollecitano un suo approfondimento.

Per questi motivi, l'ambiente nel quale si svolgono le lezioni dovrebbe assumere sempre più le caratteristiche di un *laboratorio*, nel quale si opera individualmente o in gruppo al fine di acquisire, applicare e verificare progressivamente la qualità delle proprie conoscenze e abilità, mentre se ne verifica la spendibilità nell'affrontare esercizi e problemi via via più impegnativi sotto la guida dei docenti.

Il laboratorio di matematica viene dunque inteso non come luogo fisico diverso dalla classe, ma come insieme strutturato di attività volte alla *costruzione di significati* degli oggetti matematici, in un contesto in cui il significato non risiede unicamente nello strumento utilizzato né emerge dalla sola interazione tra studente e strumento, ma si sviluppa anche dagli scopi per i quali lo strumento è usato, dai piani che vengono elaborati per usare lo strumento e dalla riflessione individuale sugli oggetti di studio e sulle attività.

Consapevoli della difficoltà di concretizzare queste considerazioni in fattive proposte didattiche, abbiamo pensato di fornire degli spunti di riflessione progettando delle attività di laboratorio, descritte nelle schede di questa dispensa e corredate da indicazioni metodologiche. Queste attività sono state progettate tenendo presente che la matematica è una “costruzione lenta e progressiva di un pensiero ricco di significato”. Il suo apprendimento deve essere attivo ma senza ridursi a un generico fare: le attività devono coinvolgere il pensiero e devono essere significanti. Devono cioè comprendere ed evidenziare la loro relazione con la realtà. E’ importante, quindi, dare rilievo al lavoro di riflessione sul linguaggio, verbale e simbolico, condizione per la riflessione più profonda sulla struttura argomentativa che caratterizza, attraverso la matematica, il pensiero scientifico. Solo toccando questo livello di questioni si esplora l’orizzonte di significato dei contenuti.

In queste schede viene inoltre privilegiata l’attività di risoluzione di problemi, poiché attraverso di essa si stimola l’inventiva e si favorisce la comprensione, e si incomincia a illuminare quella funzione di costruzione di modelli che esprime il legame profondo tra la matematica e la realtà, tra la matematica e le scienze.

Le attività di laboratorio, pur avendo un titolo che richiama allusivamente un contenuto, sono state costruite attorno ai nuclei trasversali caratteristici dei processi mentali degli allievi, quali l’argomentare, il congetturare, il dimostrare, il misurare, il risolvere e il porsi problemi, e ciò anche con il fine di contribuire allo sviluppo delle competenze matematiche richieste nelle indicazioni nazionali.

Le attività di laboratorio sono state costruite utilizzando in particolare un sistema software, STGraph, progettare per tenere presente i due aspetti dell’apprendimento di una lingua: la forma (la notazione e i concetti di base di insieme, numero, variabile, funzione, ...) e i contenuti (i teoremi che costituiscono le varie teorie).

Saper parlare (e comprendere, e leggere, e scrivere) una lingua implica una capacità produttiva: a prescindere da quali contenuti si considerano, si suppone che li si sappia esprimere, eventualmente traducendoli da o verso un’altra lingua. I contenuti esprimibili nelle lingue storico-naturali sono i più diversi, e quindi si può imparare l’inglese attraverso contenuti generici, e non solo letterari. La matematica è invece una lingua non solo specifica quanto alla forma ma anche specializzata nei contenuti, e una parte considerevole dello studio della matematica nelle scuole è dedicata all'apprendimento di contenuti di complessità crescente: insiemistica, aritmetica, algebra, geometria, trigonometria, analisi, ... D'altra parte, è come se molti studenti, impegnati su tali contenuti, perdessero di vista la componente linguistica della matematica, e quindi non acquisissero una appropriata capacità produttiva al proposito: se chiamiamo "matematichese" la componente appunto linguistica della matematica, è come se, in anni e anni di studio, gli studenti comunque non imparassero a *parlare il matematichese*, pur (eventualmente) conoscendo contenuti di matematica. E' come se per certi studenti la matematica fosse appunto solo un insieme di contenuti (e spesso anche meno: solo un insieme di tecniche, non collegate ai problemi che tali tecniche possono risolvere), e non anche uno strumento con cui "parlare di cose". E' plausibilmente questa una delle ragioni per cui certi studenti tendono a dimenticarsi in fretta i contenuti di matematica e comunque, e più criticamente, tendono a non usare metodi formali / quantitativi nel / per "parlare di cose". Si tratta dunque plausibilmente dello stesso effetto che si otterrebbe se si insegnasse in inglese la letteratura inglese a persone che non conoscono a sufficienza l'inglese, e focalizzando l'attenzione solo sugli aspetti letterari, nell'ipotesi che le questioni linguistiche possano essere date per scontate: ci si potrebbe stupire constatando che i risultati dell'apprendimento sono scarsi, e comunque prevalentemente di tipo mnemonico?

La traduzione tra lingue è un'efficace strumento di apprendimento e di verifica dell'apprendimento: ciò vale anche per le attività di traduzione dal / al matematichese. Per sviluppare o rinforzare le competenze "linguistiche matematiche" degli studenti può essere utile perciò proporre loro esercizi di "traduzione", anche molto semplici quanto ai contenuti (cosa che non dovrebbe generare barriere dovute a mancanza di pre-competenze) ma creativi / produttivi, invece che meccanici, quanto agli aspetti linguistici. La traduzione può essere certamente compiuta tra matematichese e italiano, e in particolare:

- *traduzioni dal matematichese all'italiano*, in cui si fornisce una descrizione in matematichese e si chiede di produrre in italiano una descrizione corrispondente (cioè che "spiega" in italiano la descrizione in matematichese), e che sollecitano un lavoro prevalente di analisi, per comprendere la descrizione e spiegarla;
- *traduzioni dall'italiano al matematichese*, in cui si fornisce una descrizione in italiano e si chiede di produrre in matematichese una descrizione corrispondente (cioè che "implementa" in matematichese la descrizione in italiano), e che sollecitano un lavoro prevalente di sintesi, per comprendere la descrizione e formalizzarla.

Ancora più interessanti sono però, a volte, attività di traduzione che coinvolgono una terza lingua: un linguaggio di programmazione per calcolatore, soprattutto se strutturato in modo da essere il più possibile vicino al matematichese invece che all'algoritmica. In questo modo, la focalizzazione rimane appunto sul matematichese, invece che trasferirsi all'informatica e ai problemi relativi alla struttura del calcolatore. Un appropriato ambiente software può così fungere da "laboratorio linguistico", in particolare se impiegato nella soluzione di problemi in cui l'entità in oggetto è descritta in termini dinamici e quindi l'implementazione opera come una simulazione dell'entità stessa, cosa che sollecita anche la dimensione ludica nel lavoro di traduzione.

"Perciò bisogna che ciascuno si curi in primo luogo di apprendere dalla Matematica quegli elementi insostituibili di pensiero che occorrono per penetrare e padroneggiare l'essenza dei problemi nel suo proprio campo di studi e di attività. Di fronte a tale esigenza diviene subordinata seppure non proprio secondaria quella di apprendere anche a maneggiare personalmente formule e calcoli, il cui stesso valore istruttivo e formativo dipende a sua volta dal saperli fondere in una concezione ben assimilata e bene innestata a quella delle proprie discipline preferite e alle proprie abituali abilità di ragionamento, di meditazione, di immaginazione." (Bruno de Finetti)

Schede di laboratorio

Ognuna di queste schede è strutturata in due parti: un'introduzione didattica, seguita dalla descrizione di un'attività di didattica laboratoriale, sviluppata passo passo e con considerazioni contestuali sulle competenze richieste e la modalità di presentazione.

Le attività sono proposte in una successione basata sulla logica dello sviluppo e dell'uso progressivo delle competenze, dall'impiego di sole variabili algebriche, prima scalari ("Polinomi come funzioni") e poi anche vettoriali ("Pitagora e oltre"), all'introduzione di variabili differenziali / di stato ("Stili di cambiamento"), anche in forma esplicitamente iterativa ("Radici babilonesi"). L'ultima attività ("Riassunto di numeri"), ancora con un'impostazione orientata alle competenze, propone un'introduzione ad alcuni aspetti di statistica descrittiva elementare ed è quindi più connotata anche in riferimento ai suoi contenuti.

Polinomi come funzioni

Note didattiche

Questa attività di laboratorio, in coerenza e insieme ad altre attività, vuole contribuire all'acquisizione da parte degli studenti delle seguenti competenze:

- utilizzare le tecniche e le procedure del calcolo aritmetico e algebrico, rappresentandone i risultati anche sotto forma grafica;
- adottare modelli matematici per analizzare variazioni in contesti reali e astratti e usare forme simboliche per rappresentare situazioni e strutture matematiche (in particolare comprendere modelli e relazioni di tipo funzionale);
- interpretare dati sviluppando deduzioni e ragionamenti sugli stessi anche con l'ausilio di rappresentazioni grafiche (riconoscere relazioni esistenti fra elementi e rappresentarle, distinguere le funzioni dalle relazioni e usare coordinate cartesiane, diagrammi e tabelle per rappresentare relazioni e funzioni).

Compito

- Esplorare numericamente il comportamento dei polinomi, interpretati come funzioni da numeri a numeri e dipendenti parametricamente dai coefficienti dei polinomi stessi.
- Disegnare su un piano cartesiano il grafico di un polinomio di terzo grado, in un dominio dato e per coefficienti assegnati.

Obiettivi

Oltre allo sviluppo e/o potenziamento delle competenze sopra descritte si perseguono i seguenti obiettivi:

- approfondire il livello di conoscenza dei polinomi e dell'operatività con essi, facendo in modo che gli studenti sappiano applicarli anche in contesti diversi da quelli del puro esercizio algebrico (a volte solo ripetitivo). Infatti, accade spesso, dopo una lezione teorica ben fatta del docente, che gli alunni ricordino bene le regole, ma le applichino poi meccanicamente senza averne capito il senso profondo e considerandole un puro gioco di segni senza significato. In questo modo si corre il rischio che essi acquisiscano un'immagine della matematica come una serie di ricette prive di metodo e di giustificazione. Questo spesso accade in particolare con i polinomi. Risulta quindi necessario porre l'accento sul significato delle "operazioni" che devono effettuare gli alunni, cioè sul processo piuttosto che sulla singola procedura. In altre parole occorre far capire agli studenti che i due aspetti fondamentali della matematica non vanno mai scissi: la matematica è, da un lato, uno strumento essenziale per la comprensione quantitativa della realtà, e dall'altro è sapere logicamente coerente e sistematico, caratterizzato da una forte unità culturale;
- acquisire un "pensiero funzionale" (insistendo sulla connessione fra il grafico di una funzione, l'interpretazione dell'andamento, il collegamento di questo con l'espressione algebrica della funzione e sugli aspetti numerici). La considerazione dei fenomeni a livello qualitativo deve diventare un'abitudine mentale degli alunni se si vuole fare in modo che le tecniche che l'alunno imparerà nel corso degli anni non siano oggetto di applicazione meccanica, ma frutto di riflessione sui significati nei diversi contesti proposti;
- approfondire la conoscenza del piano cartesiano e comprendere il concetto di funzione;
- esplorare numericamente il comportamento dei polinomi;
- costruire la tabella dei dati e rappresentare graficamente una semplice funzione nel piano cartesiano;
- tradurre il linguaggio comune nel linguaggio dell'algebra;
- accrescere e apprezzare il gusto della scoperta;
- imparare a generalizzare;
- imparare a presentare il lavoro svolto cogliendone gli aspetti più significativi.

Metodologia

- Didattica laboratoriale.

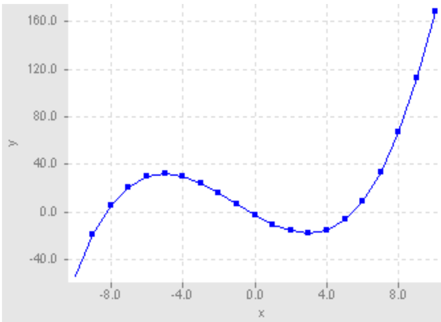
- Scoperta guidata.

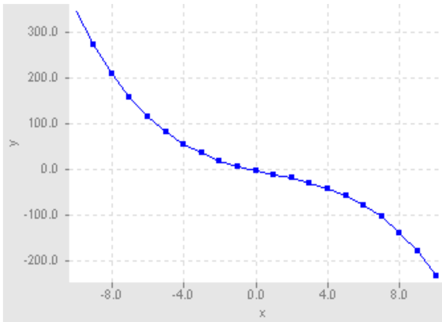
Strumenti
- STGraph.

Attività

Vogliamo esplorare qui numericamente il comportamento dei polinomi, interpretati come funzioni da numeri a numeri e dipendenti parametricamente dai coefficienti dei polinomi stessi. Il nostro obiettivo strumentale è di arrivare a disegnare su un piano cartesiano il grafico di un polinomio di terzo grado, in un dominio dato e per coefficienti assegnati.

Questi sono due esempi dei grafici che intendiamo costruire:





Supponiamo che lo strumento che possiamo usare, STGraph in questo caso, sia in grado di disegnare un segmento, e più in generale una linea spezzata, date le coordinate dei suoi vertici. Se dunque arriveremo a disporre di una lista di valori per la coordinata x e di una lista di valori corrispondenti per la coordinata y, il nostro compito potrà dirsi risolto.

Oggetto della nostra esplorazione sono i polinomi, e i polinomi di terzo grado in particolare, dunque espressioni della forma:

$$k_3x^3 + k_2x^2 + k_1x + k_0$$

in cui x è una variabile e i valori k_i sono i coefficienti del polinomio
(si potrebbe trovare qui la ragione alla base delle regole cosiddette di priorità algebrica, per cui in assenza di parentesi si calcolano prima le potenze, quindi le moltiplicazioni e infine le addizioni; in assenza di tali regole, o con regole diverse, i polinomi dovrebbero essere scritti in un modo più complicato:

$$(k_3(x^3)) + (k_2(x^2)) + (k_1x) + k_0$$

Pare effettivamente una buona ragione...).

A dispetto del fatto che questa espressione sembra forse un po' difficile da comprendere, a uno sguardo attento un polinomio si presenta semplicemente come la somma di termini tutti della stessa forma: il prodotto di una costante per la potenza della variabile
(è vero? Nel caso del primo addendo la cosa è chiara, la costante è k_3 e la potenza è x^3, e una considerazione analoga vale per il secondo addendo. Ma che dire del terzo e del quarto addendo? Sono anch'essi esprimibili in modo analogo? Possiamo trovare un'espressione generica che si adatta a tutti i termini del polinomio? Ricordando che $x^1=x$ e $x^0=1$, la risposta è semplicemente:

$$k_i x^i$$

così che il polinomio può essere anche scritto:

$$k_3x^3 + k_2x^2 + k_1x^1 + k_0x^0$$

e perciò:

$$\sum_{i=0}^{3} k_i x^i$$

Corretto?).

Cosa descrive (oppure: quale informazione porta) il nostro polinomio? Per come l'abbiamo considerato finora, si tratta solo di una variabile, benché scritta in modo un po' complesso. Infatti, se x è una variabile, lo sono anche 2x, $3x^2$, e così via.

A volte, i polinomi compaiono in espressioni della forma:

polinomio = 0

e quindi nel nostro caso:

$$k_3x^3 + k_2x^2 + k_1x + k_0 = 0$$

cosa che corrisponde a un'equazione, di terzo grado in questo caso. Il problema è allora di trovare per quali valori della variabile la condizione è soddisfatta. Lo si potrebbe chiamare il problema della ricerca degli zeri del polinomio: per quali valori di x il polinomio ha valore 0?

Ma noi vogliamo trattare qui il polinomio in modo diverso, e in effetti più generale (come vedremo, si tratta un problema dal quale il precedente può essere derivato come un caso particolare). Si parte da una constatazione molto semplice: una volta fissati i valori dei coefficienti k_i, per ogni valore di x il polinomio a sua volta assume un valore, indichiamolo con y per brevità. Se per esempio tutti i coefficienti sono uguali a 2, e quindi il polinomio è:

$2x^3 + 2x^2 + 2x + 2$

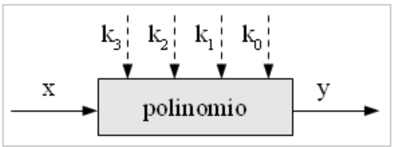
con alcuni calcoli si può costruire una tabella come questa:

x	y
0	2
1	8
2	30
…	…

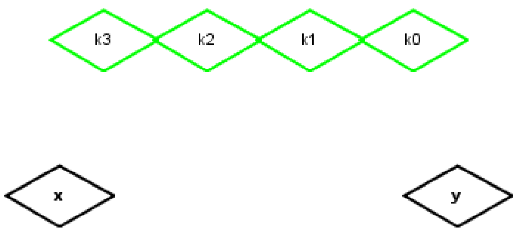
D'altra parte, se per i coefficienti si scelgono valori diversi, per esempio k_3=1, k_2=2, k_1=3, k_0=4, la tabella avrà un contenuto diverso:

x	y
0	4
1	10
2	26
…	…

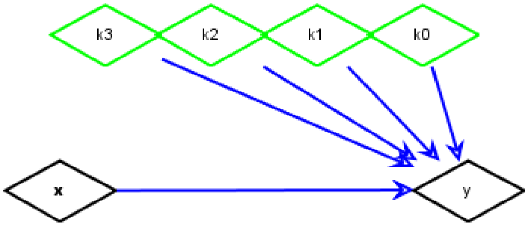
Considerato da questo punto di vista, il nostro polinomio è dunque interpretabile come una regola che, una volta stabiliti i valori dei coefficienti, stabilisce una corrispondenza tra i valori della variabile x e i valori y calcolati dal polinomio stesso. Potremmo intenderlo come un dispositivo con quattro leve di controllo, un input e un output:



Il nostro problema è dunque caratterizzato da 6 variabili, ognuna delle quali può essere rappresentata da un nodo di un modello di STGraph (coloriamo i nodi associati ai coefficienti per metterne in evidenza il ruolo appunto di "leve di controllo"):



Il passo successivo è di identificare le relazioni di dipendenza tra queste variabili, cosa che in STGraph corrisponde a definire frecce tra nodi. Abbiamo già risolto il problema: y dipende da x e dai coefficienti k_i. Dunque:

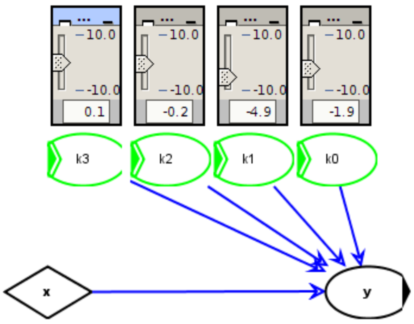


Il modello è ora completo nella sua parte qualitativa, dato che mostra quali variabili sono da considerare e come esse dipendono le une dalle altre. D’altra parte, il modello manca ancora della parte quantitativa e quindi non può essere usato per calcolare alcunché. Introduciamo, prima di tutto, la formula per y, che naturalmente non è altro che il polinomio stesso:

y ← k3*x^3+k2*x^2+k1*x+k0

(la notazione, tipica dei linguaggi di programmazione, è diversa da quella impiegata abitualmente in matematica: il simbolo per la moltiplicazione è ‘*’ – dunque 2*2 vale 4 – e il simbolo ‘^’ indica l’elevamento a potenza – dunque 3^2 vale 9).

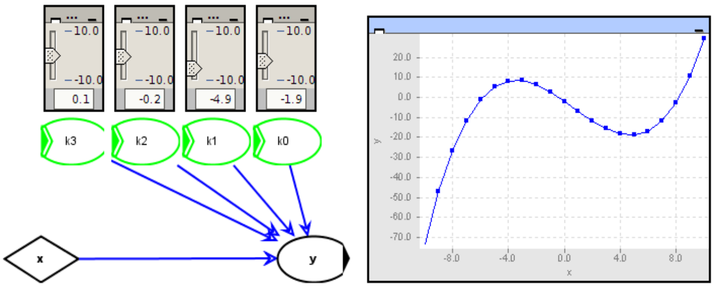
Come si vede, è proprio questa espressione che stabilisce la dipendenza quantitativa di y da x e dai coefficienti k_i. Possiamo considerare questi ultimi come i parametri del sistema di calcolo che ci apprestiamo a realizzare, e quindi ne possiamo rendere controllabili interattivamente i valori mediante oggetti “cursore” (cosa che mette ulteriormente in evidenza il ruolo di “leve di controllo” dei coefficienti). Ecco il risultato:



Rimane naturalmente da stabilire un valore per x. Come abbiamo già visto con le tabelle sopra, una volta fissati i valori dei parametri k_i per ogni valore numerico di x si ottiene un valore per y: si ottiene così una coppia di coordinate (x,y), che corrisponde a un punto su un piano cartesiano. Se ripetessimo dunque l’operazione di assegnazione di un valore per x e quindi il calcolo di y, otterremmo una lista di coppie, (x_1,y_1), (x_2,y_2), … Ma si può fare di meglio, e più facilmente, assegnando come valore di x direttamente una lista di valori, per esempio:

x ← [–10:10]

corrispondente alla lista –10, –9, …, 9, 10, e lasciando che STGraph effettui il calcolo del polinomio sull'intera lista, e quindi in parallelo su ogni elemento della lista stessa. In questo modo giungiamo a disporre di due liste di valori per x e y, esattamente quanto volevamo ottenere:



Ora che abbiamo costruito questo sistema di calcolo, possiamo sfruttarlo per esplorare sperimentalmente alcune caratteristiche del polinomio da cui siamo partiti. In particolare ci si potrebbe chiedere:

- al variare dei valori dei parametri, quanti zeri può avere il polinomio? ci sono combinazioni dei valori dei parametri per cui il polinomio non ha alcuno zero?
- che forma ha la curva nel caso in cui k_3=0? e nel caso in cui k_3=k_2=0? e nel caso in cui k_3=k_2=k_1=0?

Pitagora e oltre

Note didattiche

Questa attività di laboratorio, in coerenza e insieme ad altre attività, vuole contribuire all'acquisizione da parte degli studenti delle seguenti competenze:

- individuare le strategie appropriate per la soluzione di problemi;
- progettare e costruire modelli di situazioni reali;
- confrontare e analizzare figure geometriche, individuando invarianti e relazioni.

Compito

- Disegnare su un piano cartesiano un triangolo rettangolo generico e calcolarne la lunghezza dei lati a partire dalle coordinate dei vertici.
- Disegnare i quadrati costruiti sui lati del triangolo (livello più avanzato).

Prerequisiti

- Elementi di base di geometria analitica (piano cartesiano, coordinate dei punti).

- Enunciato del teorema di Pitagora.
- Elementi di trigonometria (per il livello più avanzato).
- Conoscenza delle tecniche di base di STGraph.

Problema didattico

Dopo una lezione teorica ben fatta del docente, accade che gli alunni ricordino bene le regole (per esempio: il quadrato costruito sull'ipotenusa è equivalente alla somma dei quadrati costruiti sui due cateti; oppure: il quadrato costruito su un cateto corrisponde alla differenza tra il quadrato costruito sull'ipotenusa e il quadrato costruito sull'altro cateto) ma mantengano un atteggiamento di applicazione meccanica e acritica delle regole stesse, che rimangono entità irrelate a fatti esperienziali e quindi, paradossalmente, strumento senza un problema per cui essere applicate.

Obiettivi

Oltre allo sviluppo e/o potenziamento delle competenze sopra descritte si perseguono i seguenti obiettivi:

- approfondire la conoscenza del teorema di Pitagora e fare in modo che gli studenti lo sappiano applicare anche in contesti diversi da quello del puro esercizio geometrico;
- esplorare numericamente qualche caratteristica dei triangoli rettangoli;
- accrescere e apprezzare il gusto della scoperta;
- imparare a generalizzare;
- imparare a presentare il lavoro svolto cogliendone gli aspetti più significativi.

Metodologia

- Didattica per problemi.
- Scoperta guidata.

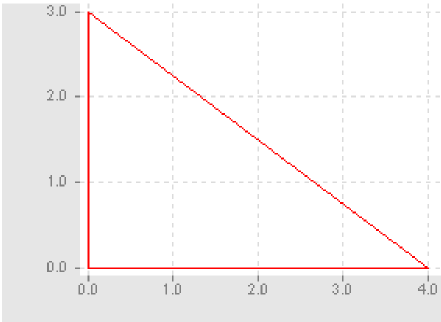
Strumenti

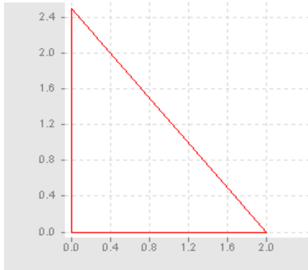
- STGraph.

Attività

Vogliamo esplorare qui numericamente qualche caratteristica dei triangoli rettangoli, in particolare in riferimento al contenuto del teorema di Pitagora. Il nostro obiettivo strumentale è di arrivare a disegnare su un piano cartesiano un triangolo rettangolo generico e a calcolarne la lunghezza dei lati, a partire dalle coordinate dei vertici. Come obiettivo più avanzato, vogliamo giungere a disegnare i quadrati costruiti sui lati del triangolo.

Questi sono due esempi dei triangoli che intendiamo costruire:

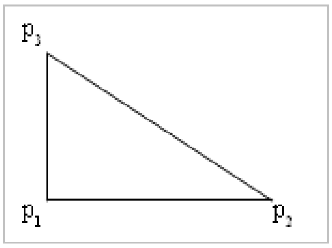




Supponiamo che lo strumento che possiamo usare, STGraph in questo caso, sia in grado di disegnare un segmento, e più in generale una linea spezzata, date le coordinate dei suoi vertici. Dunque studiamo i due triangoli proprio in riferimento alle coordinate dei loro vertici: cosa hanno in comune e quali differenze ci sono invece?

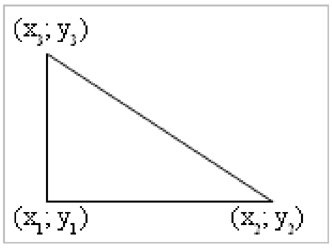
Sono due triangoli rettangoli, con il vertice all'intersezione dei due cateti nell'origine, e dunque alle coordinate (0,0; 0,0). I cateti hanno lunghezze diverse, ma in entrambi i triangoli un cateto è orizzontale e l'altro è verticale. Come possiamo esprimere dunque queste condizioni in riferimento alle coordinate dei loro vertici?

Per proseguire in modo appropriato, ed evitare di dover usare o espressioni lunghe e complicate (come "vertice all'intersezione dei due cateti") o indicazioni più o meno ambigue (come "quel vertice", puntandolo con un dito), conviene identificare univocamente ogni vertice con un nome; per esempio:

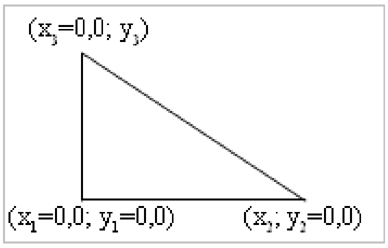


In questo modo, possiamo riesprimere appropriatamente le analogie e le differenze tra i triangoli che avevamo identificato prima: le coordinate di p_1 sono (0,0; 0,0) e l'angolo in p_1 è retto; il segmento p_1p_2 è orizzontale, di lunghezza variabile; il segmento p_1p_3 è verticale, di lunghezza variabile.

Siamo ora pronti a fare il passo successivo, assegnando dei nomi alle coordinate:

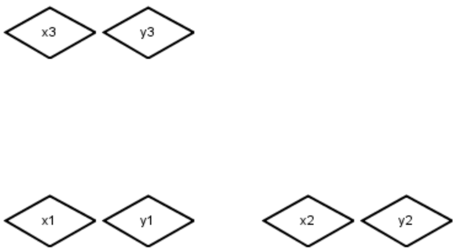


Come si possono descrivere ora le condizioni che il segmento p_1p_2 è orizzontale, il segmento p_1p_3 è verticale, ed entrambi sono di lunghezza variabile? Ecco la soluzione, direttamente nel grafico:



In sintesi, le condizioni che abbiamo identificato si traducono nella costruzione di 6 variabili, 4 delle quali in effetti con un valore pre-assegnato e da non modificare, e 2, x_2 e y_3, il cui valore può invece essere cambiato. E' chiaro, infatti, che al variare di x_2 e y_3, indipendentemente l'uno dall'altro, possiamo ottenere tutti i triangoli rettangoli che soddisfano le nostre condizioni?

Per rendere operativo tutto ciò, in un nuovo modello di STGraph introduciamo i 6 nodi corrispondenti alle coordinate dei 3 punti:



e cominciamo a costruire un determinato triangolo, per esempio quello a sinistra nella prima figura, dunque assegnando:

x1 ← 0

y1 ← 0

x2 ← 4

y2 ← 0

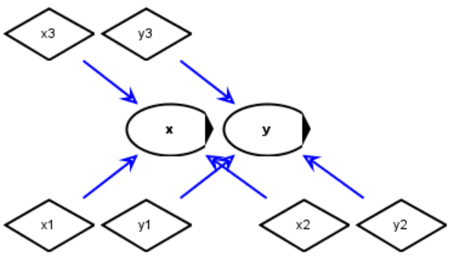
x3 ← 0

y3 ← 3

(qualche considerazione sulla notazione, e in particolare sulle differenze tra il modo di scrivere le espressioni in matematica e nei linguaggi di programmazione:

- i linguaggi di programmazione, e anche STGraph, generalmente ammettono solo una scrittura lineare (diciamo: un carattere dopo l'altro, senza alcuna formattazione), e quindi per esempio pedici, come x_1, e apici, come x^2, non si possono scrivere in questo modo: è per questo che al posto di x_1, y_1, … scriveremo d'ora in poi x1, y1, …;
- i linguaggi di programmazione, e anche STGraph, generalmente adottano il punto come separatore decimale invece della virgola, una convenzione presa dal mondo anglosassone; per semplicità, ometteremo la parte decimale dei numeri quando non rilevante, scrivendo quindi per esempio 3 invece di 3.0).

Come abbiamo accennato sopra, STGraph è in grado di disegnare una linea spezzata (e tale è il perimetro del nostro triangolo: il fatto che sia una linea chiusa non è un problema) pur di disporre di due variabili che contengano rispettivamente la lista delle coordinate x e la lista delle coordinate y dei vertici; introduciamo dunque i nodi x e y, e quindi le frecce che descrivono le relazioni di dipendenza: il valore di x dipende da quello di x1, x2 e x3, e quindi ci dovranno essere frecce che da x1, x2 e x3 vanno a x, e la stessa cosa per i nodi relativi alle coordinate y:



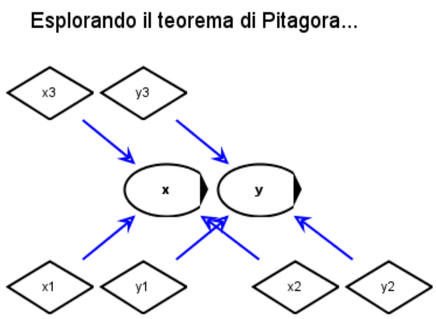
(notate la diversa forma dei nodi x e y rispetto a quelli associati alle coordinate: STGraph riconosce la differenza tra nodi con valore costante – disegnati come rombi – e nodi il cui valore è calcolato in funzione di altri nodi – e questi sono disegnati come forma ellittica).

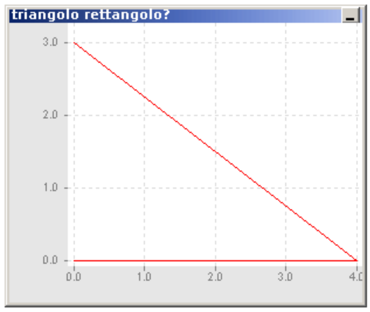
Definiamo ora le espressioni che costruiscono le due liste: è questa la soluzione corretta?

x ← [x1, x2, x3]

y ← [y1, y2, y3]

(notate che in queste espressioni la virgola è il separatore tra elementi della lista, e non il separatore decimale). Mettiamola alla prova facendo disegnare a STGraph la serie (x; y) in un oggetto “grafico”. Questo è il risultato:





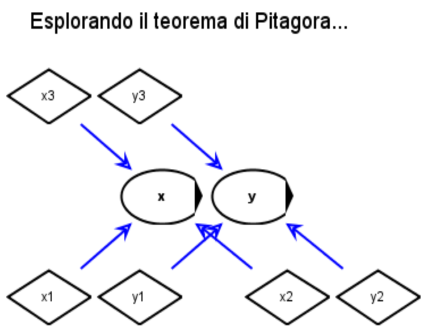
Evidentemente c’è qualcosa che non va: la linea spezzata non è chiusa, e quindi non può rappresentare il perimetro di una figura geometrica. E’ chiaro dov’è l’errore che abbiamo commesso?

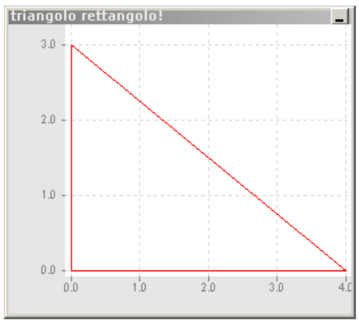
La soluzione corretta è dunque:

x ← [x1, x2, x3, x1]

y ← [y1, y2, y3, y1]

e in effetti ridisegnando:

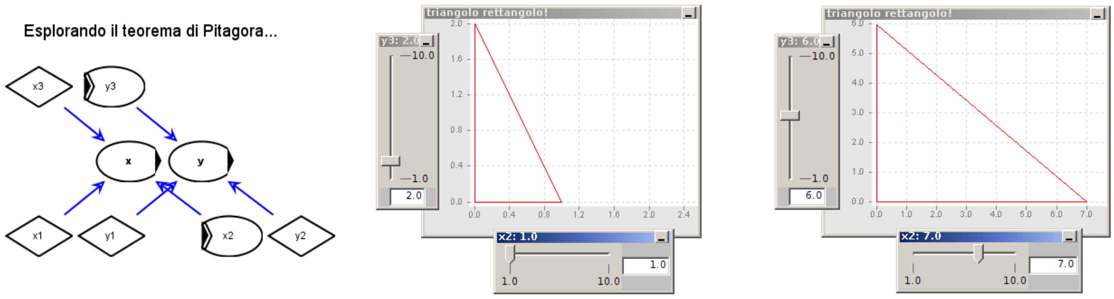




Proprio quello che volevamo ottenere!

Come prossimo passo, recuperiamo un’informazione che finora è rimasta inutilizzata: i nodi corrispondenti alle coordinate x2 e y3 non dovrebbero essere descritti come costanti, dato che appunto al variare del loro valore ci consentono di generare i diversi triangoli rettangoli.

Associamo perciò a ciascuno di essi un oggetto “cursore”, così che eseguendo la simulazione possiamo modificare interattivamente i valori delle due variabili e vedere ogni volta un nuovo triangolo disegnato. Ecco due esempi (notate che i nodi x2 e y3 hanno cambiato forma: ora sono ellissi con una freccia entrante, a indicare che rappresentano variabili il cui valore è assegnato direttamente da noi; sono, come si dice, variabili “di input”):



Siamo pronti ora per usare il teorema di Pitagora, che qui ci serve per risolvere il problema di calcolare la lunghezza dei lati di ognuno dei triangoli che possiamo costruire.

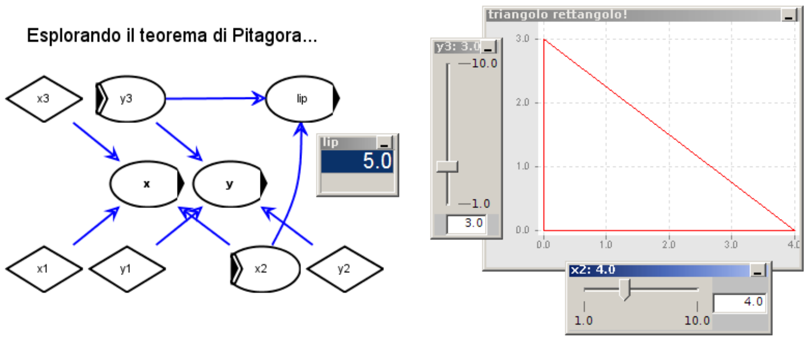
Dobbiamo partire però dalla lunghezza dei cateti: come la si trova?

Dato che abbiamo assunto la condizione che un cateto sia orizzontale e uno sia verticale, la soluzione è semplice: le due lunghezze sono x2–x1 e y3–y1. Ancora meglio: poiché, sempre per le condizioni da cui siamo partiti, x1=y1=0, le lunghezze sono proprio x2 e y3 ! Aggiungiamo allora al nostro modello un nuovo nodo – chiamiamolo per esempio lip, come “lunghezza ipotenusa” – e applichiamo il teorema di Pitagora per calcolarne il valore:

lip ← sqrt(x2^2 + y3^2)

(la notazione appare un poco inusuale: è tipica dei linguaggi di programmazione, che generalmente non sono in grado di impiegare simboli speciali come quello per la radice quadrata e devono compensare con nomi di funzioni – spesso derivati dall’inglese, come “sqrt” (*square root*) – e parentesi; inoltre, la possibilità di sola scrittura lineare, come già considerato in precedenza, impone che gli esponenti delle potenze non possano essere scritti come apici, da cui la scrittura a^b per indicare “a elevato alla b”).

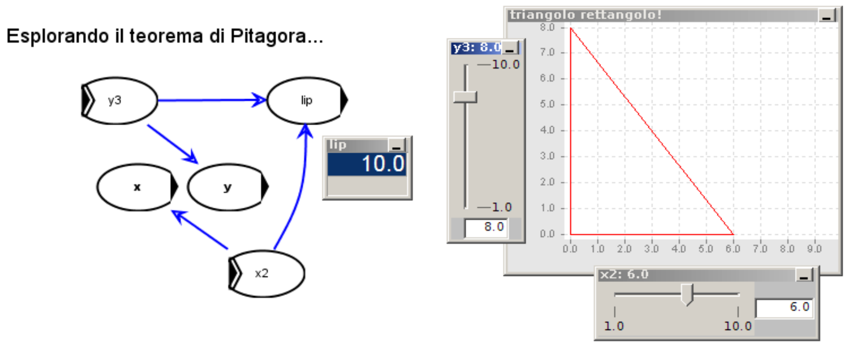
Ecco il risultato, con un oggetto “tabella di visualizzazione” per presentare il valore calcolato di lip:



Per finire (questa parte del lavoro), potremmo decidere di semplificare il grafo, eliminando i nodi costanti: le condizioni che abbiamo adottato ci impongono infatti che essi abbiano sempre valore 0, che quindi può essere scritto direttamente nelle liste di x e y, che in questo modo diventano:

x ← [0, x2, 0, 0]

y ← [0, 0, y3, 0]



Abbiamo con ciò creato un "calcolatore dedicato" a disegnare interattivamente particolari triangoli rettangoli e a calcolarne la lunghezza dell'ipotenusa (questa attività potrebbe essere replicata per esempio a proposito di triangoli isosceli).

Giungere a disegnare i quadrati costruiti sui cateti è un compito non più difficile di quanto realizzato finora.

Per ogni cateto, ci bastano due nodi per mantenere la lista delle coordinate x e y dei vertici del quadrato corrispondente. Da quali altri nodi tali liste dipendono, e quindi quali frecce dovremo costruire?

Sempre per le condizioni che abbiamo adottato, tutte le coordinate relative al cateto orizzontale sono pari a 0 o a x2, e la stessa cosa vale per il cateto verticale e y3. E come sono definite queste liste?

Ecco la soluzione:

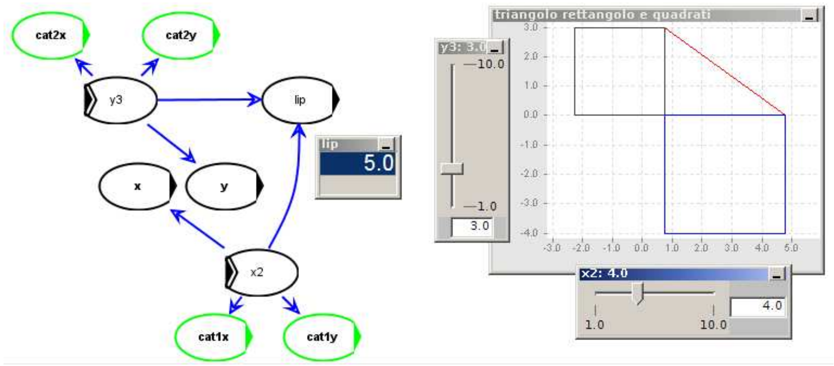
cat1x ← [0, x2, x2, 0, 0]

cat1y ← [0, 0, –x2, –x2, 0]

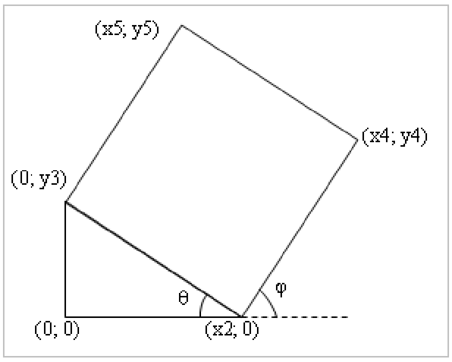
cat2x ← [0, 0, –y3, –y3, 0]

cat2y ← [0, y3, y3, 0, 0]

ed ecco il risultato:



Per disegnare il quadrato costruito sull'ipotenusa occorre invece un po' di trigonometria. Cominciamo a fare "a mano" un disegno appropriato:



Da qui si vede che ci occorre calcolare l'ampiezza dell'angolo φ, che a sua volta dipende dall'ampiezza dell'angolo θ. Poiché:

x2 = lip * cos(θ)

si ha che:

θ ← acos(x2 / lip)

e dal fatto che:

$\varphi = \pi - (\theta + \pi / 2)$

otteniamo finalmente che:

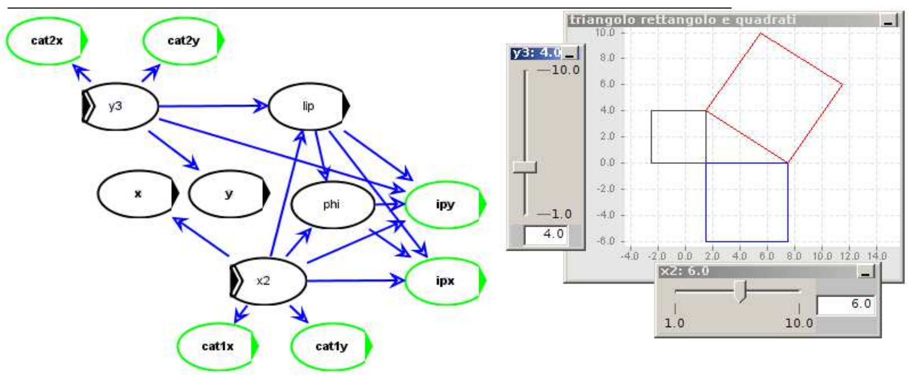
$\varphi \leftarrow \pi / 2 - \theta = \pi / 2 - \text{acos}(x2 / \text{lip})$

A questo punto possiamo costruire direttamente le liste con le coordinate x e y del quadrato:

ipx ← [x2, x2+lip*cos(phi), lip*cos(phi), 0, x2]

ipy ← [0, lip*sin(phi), y3+lip*sin(phi), y3, 0]

Una volta che siano stati definiti questi nodi, il modello si presenta così:



Il nostro problema è così completamente risolto.

Stili di cambiamento

Note didattiche

Questa attività di laboratorio, in coerenza e insieme ad altre attività, concorre al raggiungimento della seguente finalità:

- identificare e comprendere il ruolo che la matematica gioca nel mondo reale;
- utilizzare la matematica e confrontarsi con essa in modi che rispondono alle esigenze della vita di un cittadino che sappia esercitare un ruolo costruttivo, impegnato e basato sulla riflessione,

e vuole contribuire all'acquisizione da parte degli studenti delle seguenti competenze:

- adottare modelli matematici per analizzare variazioni in contesti reali e astratti e usare forme simboliche per rappresentare situazioni e strutture matematiche;
- operare valutazioni ben fondate;
- individuare le strategie appropriate per la risoluzione dei problemi;

- interpretare dati sviluppando deduzioni e ragionamenti sugli stessi anche con l'ausilio di rappresentazioni grafiche, usando consapevolmente gli strumenti di calcolo;
- operare collegamenti tra gli elementi matematici e quelli fisici.

Compito

- Esplorare numericamente qualche aspetto elementare del concetto di variazione di una grandezza, rilevante in particolare in tutti i fenomeni caratterizzati da una o più variabili dipendenti dal tempo.
- Disegnare su un piano cartesiano la traiettoria definita dai valori della grandezza in un intervallo di tempo.

Obiettivi

Oltre allo sviluppo e/o potenziamento delle competenze sopra descritte, si perseguono i seguenti obiettivi:

- applicare i concetti appresi in un dato contesto a un altro contesto, comprendendo il significato simbolico dei termini matematici;
- collegare i concetti della matematica a quelli della fisica;
- usare opportune schematizzazioni matematiche per descrivere e interpretare situazioni e fenomeni, e formalizzare il percorso di soluzione di un problema mediante modelli matematici di varia natura e attraverso la sua strutturazione in tappe;
- tradurre dal linguaggio naturale al linguaggio matematico e viceversa, distinguendo, in particolare, le variabili e le costanti;
- fare previsioni;
- operare simulazioni;
- convalidare i risultati conseguiti sia empiricamente che mediante argomentazioni;
- accrescere e apprezzare il gusto della scoperta;
- imparare a generalizzare.

Metodologia

- Problem solving.
- Didattica laboratoriale.
- Case study.
- Scoperta guidata.

Strumenti

- STGraph.

Attività

Vogliamo esplorare qui numericamente qualche aspetto elementare del concetto di variazione di una grandezza, rilevante in particolare in tutti i fenomeni caratterizzati da una o più variabili dipendenti dal tempo. Il nostro obiettivo strumentale è di arrivare a disegnare su un piano cartesiano la traiettoria definita dai valori della grandezza in un intervallo di tempo, in modo da poter studiare visualmente la relazione tra il criterio di variazione e la forma della traiettoria.

Uno dei concetti introduttivi della fisica è quello di velocità. Come conoscenza preliminare è sufficiente assumere che sia data una grandezza "posizione nello spazio" di un oggetto, indichiamola con p, e che si riconosca che tale posizione può cambiare nel corso del tempo, così che p(t) rappresenta la posizione dell'oggetto all'istante t. Se in istanti diversi, t_1 e t_2, la posizione è a sua volta diversa, cioè $p(t_1) \neq p(t_2)$, è evidentemente perché l'oggetto nel frattempo si è spostato. A parità di durata dell'intervallo di tempo, $t_2 - t_1$, maggiore è la distanza tra $p(t_1)$ e $p(t_2)$ e più velocemente dunque deve essersi spostato l'oggetto. La grandezza "velocità nello spazio" è definita allora come:

$$v(t_1) = \frac{p(t_2) - p(t_1)}{t_2 - t_1}$$

una formula il cui significato è: se l'oggetto all'istante t_1 si trova nella posizione $p(t_1)$, per raggiungere la posizione $p(t_2)$ dopo un tempo $t_2 - t_1$ deve spostarsi con la velocità $v(t_1)$. D'altra parte, per come è scritta, la formula è proprio facilmente interpretabile: come possiamo renderla più chiara?

A questo scopo facciamo due piccole modifiche:

- mettiamo meglio in evidenza che stiamo compiendo un'osservazione nel corso di un intervallo che ha una certa durata; se scriviamo tale durata come:

  $\Delta t = t_2 - t_1$

  la formula può essere riscritta in questo modo:

  $$v(t_1) = \frac{p(t_1 + \Delta t) - p(t_1)}{\Delta t}$$

  cosa che ci consente tra l'altro di eliminare l'indice:

  $$v(t) = \frac{p(t + \Delta t) - p(t)}{\Delta t}$$

  semplificando ancora di più le cose;

- mettiamo meglio in evidenza che la velocità è lo strumento del cambiamento, ma che noi siamo (generalmente) interessati a studiare la posizione; in due passaggi di algebra elementare la nostra formula diventa:

$$p(t + \Delta t) = p(t) + v(t)\,\Delta t$$

Ora è finalmente un po' più esplicito il suo significato, soprattutto se intendiamo che l'istante t sia il presente e t + Δt sia un istante futuro (probabilmente non tanto lontano, a meno che non scegliamo un Δt grande). Possiamo leggerla così: in ogni istante, se conosco la posizione e la velocità dell'oggetto posso calcolare, e quindi prevedere, la posizione che l'oggetto stesso avrà in un istante futuro. Così espressa, quella che sembrava solo la definizione della velocità è diventata addirittura uno strumento di previsione!

Potrebbe essere interessante allora metterla alla prova, magari affidandoci alla fantasia a proposito degli oggetti i cui cambiamenti cerchiamo di descrivere. In fondo, pur di accettare una certa generalizzazione, "oggetti" che "si spostano" non si trovano certo solo in fisica. Per esempio, il nostro "oggetto" potrebbe essere un nuovo modello di telefono cellulare, la cui "posizione" all'istante (supponiamo: nella settimana) t è il numero complessivo di esemplari del telefono posseduti dai clienti in una certa città. E così, sempre per esempio:

settimana, t	numero di telefoni, p(t)
1	100
2	350
3	650
...	...

Nel corso delle prime due settimane la variazione è stata di 350 – 100 = 250 e di 650 – 350 = 300 telefoni rispettivamente, così che, trattando proprio la settimana come unità di misura, possiamo scrivere:

v(1) = 250 telefoni/settimana

v(2) = 300 telefoni/settimana

Dunque non solo il telefono si sta diffondendo, ma addirittura la velocità con cui si diffonde nel mercato sta crescendo. Ma guarda... La velocità a sua volta sta cambiando: non potremmo allora chiederci con che velocità la velocità cambia?

Facciamo un passo per volta, e torniamo per ora alla nostra formula di previsione con qualche problema:

(a) se per le prime 10 settimane dal lancio del telefono sul mercato ogni settimana si sono venduti 100 telefoni, quanti telefoni ci saranno alla fine?

(b) e se invece in ognuna delle 10 settimane, grazie al passa parola, ogni 4 persone che avevano il telefono altri 3 l'hanno acquistato?

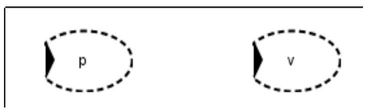
(c) e se invece, grazie agli investimenti in pubblicità dell'azienda produttrice, in ognuna delle 10 settimane il numero di telefoni venduti è aumentato di 20 unità rispetto al numero di telefoni venduti nella settimana precedente?

Questi problemi condividono una stessa struttura: ci richiedono di fare una previsione sulla "posizione" che avrà un "oggetto" in un momento futuro a partire da condizioni specificate, e per tutti e tre la formula:

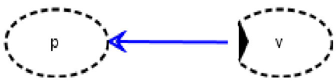
$$p(t + \Delta t) = p(t) + v(t)\, \Delta t$$

è lo strumento di base che può essere usato.

In questo caso (pochi e piccoli numeri) potremmo naturalmente fare i conti "a mano", ma proviamo invece a lavorare con STGraph. A questo proposito il primo passo, dopo aver creato un nuovo modello, è di identificare le variabili rilevanti, in questo caso un compito facile: p e v. Introduciamole come nodi nel grafo:



Occorre ora stabilire le relazioni di dipendenza tra variabili: per ora su v non poniamo vincoli, e quindi lasciamola senza dipendenze, mentre chiaramente p dipende da v:



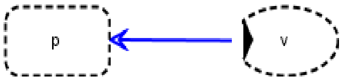
D'altra parte, p dipende anche da se stessa, e più precisamente dal valore che essa aveva un Δt (nei nostri esempi è una settimana) prima. Tutte le volte che ciò accade (e se ricordiamo come siamo giunti a questa formula, tutte le volte dunque che è in gioco una velocità), cioè quando una variabile viene calcolata non sincronicamente rispetto alle variabili da cui dipende:

$$x(t) = f(t)$$

ma un Δt successivo:

$$x(t + \Delta t) = f(t)$$

diciamo che la variabile in questione è "di stato", invece che "algebrica". Modificando il tipo della variabile, il nodo corrispondente cambia automaticamente forma:



In STGraph, ogni variabile di stato ha tre caratteristiche di base:

- il suo valore viene calcolato per il Δt successivo (per ragioni tecniche l'espressione in questo caso si chiama "funzione di transizione di stato");
- può dipendere da se stessa, e il suo valore attuale è rappresentato dalla variabile "this", che potremmo dunque interpretare come "il mio valore in questo istante";
- può essere calcolata solo se è dato il valore che essa assume in un "istante iniziale", t_0, valore che è generalmente chiamato "stato iniziale".

Se ipotizziamo che nel nostro caso lo stato iniziale sia 0 (cioè la simulazione parte da una condizione in cui non sono ancora stati venduti telefoni), la definizione di p si scrive:

p-init ← 0

p-trans ← this + v * timeD

("timeD" è la variabile di STGraph corrispondente a Δt; se, come abbiamo considerato, assumiamo la settimana come unità di misura, allora timeD è pari a 1 e quindi possiamo tralasciarlo come fattore moltiplicativo nelle formule).

Il problema (a) è il più semplice: il fatto che in ogni settimana si vendono 100 telefoni, corrisponde a:

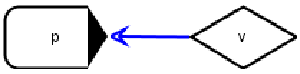
p-trans ← this + 100

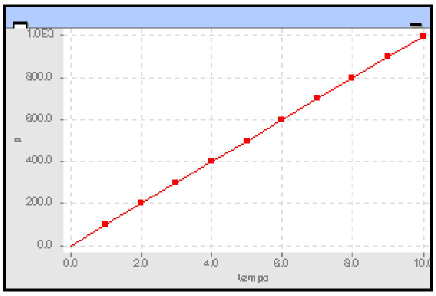
cioè:

v ← 100

p-trans ← this + v

Ecco il risultato (piuttosto ovvio...) della simulazione:





tempo p

tempo	p
0	0
1	100
2	200
3	300
4	400
5	500
6	600
7	700
8	800
9	900
10	1000

Tornando per un attimo alla fisica, e interpretando perciò le variabili p e v con il loro usuale significato meccanico di posizione e velocità, possiamo notare che il problema (a) presenta dunque un caso di moto rettilineo uniforme, appunto caratterizzato da velocità costante e quindi moto inerziale (un cenno circa una possibile interpretazione dinamica, e non solo cinematica, del problema: il moto è inerziale nel caso in cui non ci sono forze esterne oppure, più in generale, quando esse sono nulle: dato che nessun prodotto industriale "si vende da solo", possiamo immaginare che un numero costante di telefoni venduti al mese corrisponda a una situazione in cui l'"attrito", cioè la difficoltà a vendere, è compensato dalla "forza" di vendita).

Prendiamo ora in esame il problema (b): in questo caso, la variazione della "posizione" è descritta dalla condizione: ogni 4 possessori del telefono nel mese corrente altri 3 lo acquisteranno nel corso del mese stesso. Come nel caso precedente, si tratta di una condizione a proposito della definizione della "velocità", che dunque ora dipende dalla posizione. Dobbiamo perciò modificare la struttura del grafo, in modo da descrivere questa dipendenza reciproca (la posizione dipende dalla velocità, ma la velocità dipende dalla posizione):

Ma è possibile ciò? Se le due variabili implicate fossero entrambe algebriche la risposta sarebbe certamente negativa. Se infatti, per esempio, definissimo il sistema:

$$\begin{cases} p(t) = 2v(t) \\ v(t) = p(t) + 1 \end{cases}$$

corrispondente alle istruzioni di assegnazione:

$p \leftarrow 2 * v$

$v \leftarrow p + 1$

in ogni istante t per calcolare p(t) dovremmo conoscere il valore di v(t), ma per calcolare v(t) dovremmo conoscere il valore di p(t), cosa che chiaramente genererebbe un regresso all'infinito.

Il punto essenziale in questo caso è però che una delle due variabili, p, non è algebrica ma di stato, e quindi il suo valore è calcolato "per un istante dopo". Quindi possiamo proseguire: come deve essere definita la variabile v in questo caso? Vediamo qualche caso: se p(t) = 4 allora p(t + Δt) = 4 + 3, e poiché in generale p(t + Δt) = p(t) + v(t) se ne conclude che se p(t) = 4 allora v(t) = 3. Con un po' di riflessione arriviamo alla formula:

$$v(t) = \frac{3}{4}p(t)$$

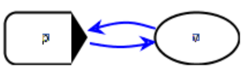
e quindi:

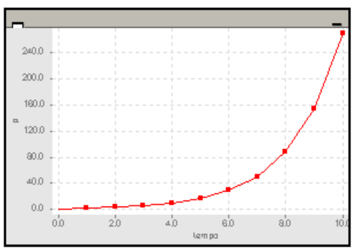
v ← 3 * p / 4

Con ciò non siamo però ancora pronti a realizzare la nuova simulazione. C'è prima di tutto un problema da risolvere: se mantenessimo, come prima, il valore 0 per lo stato iniziale la velocità iniziale sarebbe a sua volta nulla e "il sistema non partirebbe mai". Dunque per evitare questa situazione banale in questo caso dobbiamo assumere uno stato iniziale diverso da zero. Supponiamo che sia:

p-init ← 1

(all'inizio del periodo di osservazione / simulazione nel mercato c'è un telefono). E qui si presenta un secondo problema. Se p(0) = 1, allora v(0) = 0,75, cosa che implicherebbe che p(2) = 1,75. Un telefono e tre quarti nel mercato? Chiaramente la cosa non ha senso! Possiamo però decidere di disinteressarci della questione, per esempio assumendo che, tutte le volte che servirà, applicheremo un opportuno arrotondamento ai risultati del calcolo. Ecco il risultato (la colonna di destra della tabella contiene proprio i valori di "posizione" arrotondati):





tempo p p

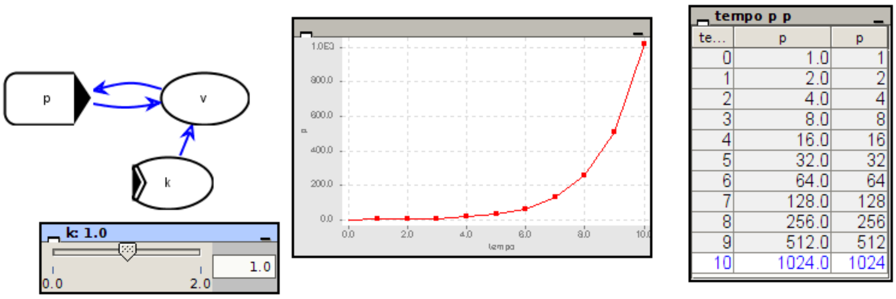
te...	p	p
0	1.0	1
1	1.75	2
2	3.0625	3
3	5.3594	5
4	9.3789	9
5	16.4131	16
6	28.7229	29
7	50.2651	50
8	87.9639	88
9	153.9368	154
10	269.3894	269

Anche a proposito di questa soluzione possiamo cercare una possibile interpretazione alternativa, ma per diversificare questa volta non rivolgiamoci alla fisica. Un sistema come:

$$\begin{cases} p(t + \Delta t) = p(t) + v(t)\Delta t \\ v(t) = kp(t) \end{cases}$$

con k costante (nel nostro caso k = 0,75), descrive per esempio il numero di individui di una popolazione che in ogni unità di tempo cresce con un tasso pari a k (cioè, se lo si considera più facile da comprendere, con un tasso del 100k%; nel nostro caso del 75%), cosa che fa sì che in questo caso la velocità corrisponde al numero di nuovi nati (un'altra interpretazione si ottiene considerando come posizione la quantità di denaro depositata in un conto bancario: k descrive allora il tasso di interesse per unità di tempo, e la velocità corrisponde agli interessi che sono maturati nell'unità di tempo e a loro volta depositati nel conto).

Potremmo poi generalizzare il modello precedente, mettendo in evidenza il tasso di crescita e introducendo l'opzione di modificarlo, per sperimentare le variazioni della "posizione" al variare di k:



te...	p	p
0	1.0	1
1	2.0	2
2	4.0	4
3	8.0	8
4	16.0	16
5	32.0	32
6	64.0	64
7	128.0	128
8	256.0	256
9	512.0	512
10	1024.0	1024

(la figura presenta la situazione in cui il tasso di crescita è del 100%, corrispondente alla condizione in cui la "posizione" raddoppia il suo valore ad ogni unità di tempo).

Siamo pronti per affrontare il problema (c), il più complesso. Ricordiamo che in questo caso il numero di telefoni venduti nella settimana t + Δt è superiore di 20 unità rispetto al numero di telefoni venduti nella settimana t. Attenzione: stiamo parlando di non telefoni complessivamente presenti nel mercato (la "posizione"), ma appunto di telefoni venduti nel corso della settimana, variabile che, come abbiamo visto, corrisponde alla "velocità" all'istante t. Ciò significa che la condizione in questo caso è:

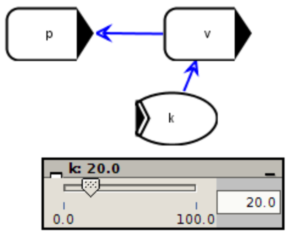
$$v(t + \Delta t) = v(t) + 20$$

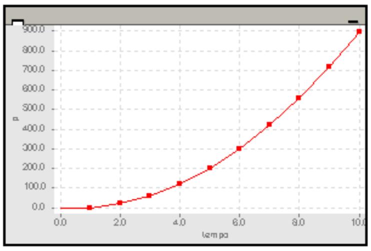
Abbiamo ormai imparato che da una condizione di questo genere dobbiamo concludere che anche v è una variabile di stato. Supponendo che lo stato iniziale sia 0, possiamo "tradurre" così:

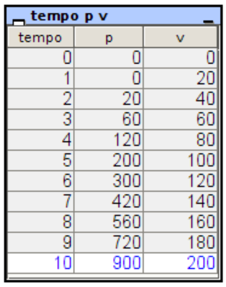
v-init ← 0

v-trans ← this + 20

Ed ecco il risultato (per rendere i conti più comprensibili, e semplici da controllare, la tabella contiene una colonna per la velocità):







tempo p v

tempo	p	v
0	0	0
1	0	20
2	20	40
3	60	60
4	120	80
5	200	100
6	300	120
7	420	140
8	560	160
9	720	180
10	900	200

Con quest'ultimo caso torniamo alla fisica. Il numero 20 rappresenta la variazione, costante dunque, della velocità. Ma come si chiama in fisica la grandezza che descrive la variazione della velocità? Accelerazione. E infatti la formula:

$$v(t + \Delta t) = v(t) + 20$$

è chiaramente un caso particolare di:

$$a(t) = \frac{v(t+\Delta t)-v(t)}{\Delta t}$$

in cui con a(t) indichiamo appunto l'accelerazione al tempo t, e in questo caso a(t) = 20. Dunque questo è un caso di moto ad accelerazione costante: come si chiama in fisica? Moto uniformemente accelerato. Proprio come quello di un corpo in un campo gravitazionale. Insomma: che ce ne fossimo resi conto o meno, il problema (c) è appunto un problema... di caduta libera (chi volesse provare ora a giocare un po' con le formule, potrebbe prendere in esame il sistema:

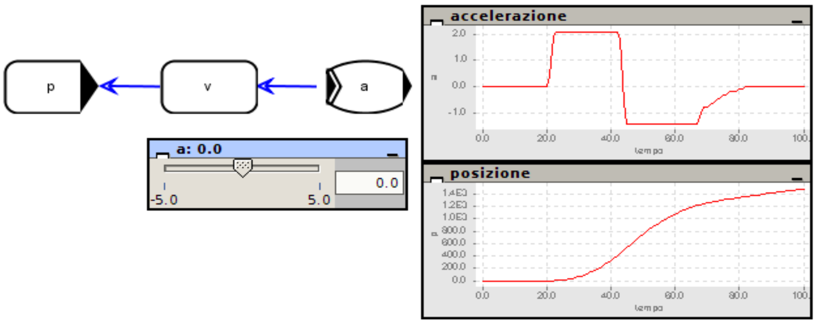
$$\begin{cases} p(t+\Delta t) = p(t) + v(t)\Delta t \\ v(t+\Delta t) = v(t) + a(t)\Delta t \end{cases}$$

nel caso in cui a(t) = k e gli stati iniziali p(0) e v(0) siano dati, per cercare di risolvere il sistema, cioè per trovare l'espressione che consente di calcolare p(n Δt) per n>0 qualsiasi. Ciò consentirebbe di calcolare direttamente la previsione per un qualsiasi istante futuro. Con qualche passaggio algebrico si ottiene:

$$p(n\,\Delta t) = p(0) + n\,\Delta t\, v(0) + \frac{1}{2}k\,(n\,\Delta t)^2$$

che, a meno della sostituzione n $\Delta t \to t$, è l'equazione cinematica del moto uniformemente accelerato: quell'equazione è dunque la "soluzione analitica" del sistema, che STGraph calcola implicitamente per noi nel corso della simulazione).

Un'ulteriore generalizzazione è, a questo punto, a portata di mano: cosa succederebbe reinterpretando il nodo k per quello che in effetti è, cioè un'accelerazione, e quindi lasciando a chi gioca con il nostro simulatore la libertà di cambiare il valore di tale variabile durante la simulazione stessa? Ecco un esempio:



Per comprendere il significato di questo "esperimento" possiamo immaginare che p(t) descriva la posizione di un oggetto che si muove senza attrito lungo una retta (certo! parlando di "posizione" finora abbiamo sempre implicitamente assunto uno spazio unidimensionale: un'interessante estensione del nostro progetto potrebbe essere di passare a descrivere moti bidimensionali), per esempio un missile nello spazio, per cui accelerazione positiva corrisponde a motori accesi in un verso e accelerazione negativa a motori accesi nell'altro verso. La figura mostra allora un esempio: dopo una prima fase in cui i motori sono spenti (a = 0) e la posizione non cambia (p = 0, lo stato iniziale), intorno a t = 20 abbiamo acceso brevemente gli avan-motori, e quindi la posizione comincia a crescere; per arrestare il moto non ci sono freni nello spazio, e perciò intorno a t = 40 abbiamo acceso i retro-motori; mano a mano che la posizione si è stabilizzata abbiamo ridotto l'accelerazione negativa.

Da qui molti altri "esperimenti" potrebbero essere progettati e realizzati. Ma questa è un'altra storia.

Radici babilonesi

Note didattiche

Questa attività di laboratorio, in coerenza e insieme ad altre attività, vuole contribuire all'acquisizione da parte degli studenti delle seguenti competenze:

- individuare le strategie appropriate per la soluzione di problemi;

- progettare e costruire modelli di situazioni reali;
- utilizzare tecniche e procedure di calcolo aritmetico e algebrico.

Compito

- Individuare un algoritmo risolutivo per calcolare, in modo approssimato, le radici quadrate di numeri positivi.
- Implementare l’algoritmo al calcolatore.
- Costruire un modello in grado di realizzare lo stesso calcolo in forma iterativa, mediante variabili di tipo differenziale (livello più avanzato).

Prerequisiti

- Elementi di calcolo algebrico.
- Conoscenza dei radicali.
- Enunciato del teorema di Pitagora.
- Concetto di algoritmo.
- Conoscenza delle tecniche di base di STGraph.

Obiettivi

- In particolare delle competenze individuate in questa attività si svilupperanno e/o potenzieranno le seguenti abilità:
- utilizzare il linguaggio simbolico;
- tradurre dal linguaggio naturale al linguaggio algebrico e viceversa;
- progettare un percorso risolutivo strutturato in tappe;
- riconoscere i dati iniziali e individuare gli obiettivi;
- schematizzare in passi elementari il procedimento risolutivo;
- formalizzare il percorso risolutivo attraverso un modello;
- controllare le procedure e verificare la rispondenza del risultato alle richieste;
- argomentare i risultati

Inoltre si individuano anche i seguenti obiettivi:

- approfondire il livello di conoscenza dei radicali puntando l’attenzione sulle procedure applicate e non sul calcolo ”bruto” utilizzando,in questo modo,consapevolmente le tecniche e le abilità di calcolo;
- comprendere il significato della formalizzazione;
- accrescere ed apprezzare il gusto della scoperta;
- imparare a generalizzare;
- imparare a presentare il lavoro svolto cogliendone gli aspetti più significativi;
- fornire semplici elementi di storia della matematica (matematica babilonese).

Metodologia

- Lezione partecipata.
- Didattica per problemi.
- Scoperta guidata.

Strumenti

- STGraph.

Attività

Vogliamo esplorare qui un algoritmo, probabilmente già noto ai babilonesi, per il calcolo approssimato di radici quadrate, studiando come trasformare un semplice concetto di matematica elementare in una procedura di calcolo, attuando dunque un processo di traduzione di una descrizione espressa in italiano in una successione di funzioni. Il nostro obiettivo strumentale è di arrivare a calcolare in modo approssimato radici quadrate di generici numeri positivi. Come obiettivo più avanzato, vogliamo giungere a costruire un modello in grado di realizzare lo stesso calcolo in forma iterativa, mediante variabili di tipo differenziale.

Il calcolo di radici quadrate è spesso utile, per esempio per trovare la lunghezza del lato di un quadrato di cui è nota l'area o, analogamente, per applicare il teorema di Pitagora. Salvo che per i cosiddetti "quadrati perfetti", come 4, 9, 16, …, calcolare la radice quadrata di un numero è considerato un procedimento complesso, che deleghiamo generalmente a calcolatrici o a programmi per calcolatori. D'altra parte, esiste una tecnica per calcolare radici quadrate che pare impiegassero già i babilonesi, notoriamente sprovvisti di strumenti elettronici. E' facile spiegare come funziona, e non è difficile trasformarla in un'insieme di istruzioni elementari, cioè in un algoritmo, che potremo eseguire direttamente oppure anche "implementare" in un programma per calcolatore.

Introduciamo il nostro problema con un minimo di notazione. Se x è un numero positivo, la sua radice quadrata, $\sqrt{x}$, è un numero y tale che y*y = x. Il nostro dato di partenza è dunque un valore x>0, e il nostro obiettivo è di trovare quel valore y tale che y=$\sqrt{x}$.

Dato che non conosciamo già y (altrimenti il problema sarebbe con ciò risolto), scegliamo un valore y_1, con la sola condizione che esso stesso sia maggiore di 0, da considerare come il primo candidato per $\sqrt{x}$. Dato che si tratta solo appunto di una prima ipotesi, possiamo tranquillamente tirare a indovinare, oppure possiamo scegliere, per esempio y_1=x/2.

Consideriamo ora il valore z_1 tale che y_1*z_1 = z, cioè z_1 = z/y_1: se y_1 e z_1 fossero uguali, il nostro problema sarebbe risolto; altrimenti (come è probabile che sia), y_1 può essere o minore o maggiore dell'ancora ignoto $\sqrt{x}$. Se $y_1<\sqrt{x}$, allora $z_1>\sqrt{x}$; viceversa, se $y_1>\sqrt{x}$, allora $z_1<\sqrt{x}$. Dunque in entrambi i casi $\sqrt{x}$sarà maggiore del minimo tra y_1 e z_1 e minore del massimo tra essi. Se come secondo candidato prendiamo il valor medio di y_1 e z_1 $y_2=(y_1+z_1)/2$, possiamo perciò supporre di migliorare la nostra scelta.

A questo punto si può ricominciare, calcolando $z_2 = x/y_2$ e verificando se y_2 e z_2 non sono uguali, così da prendere il loro valor medio, $y_3=(y_2+z_2)/2$, e così via fino alla ripetizione n-esima, in cui y_n e z_n sono almeno "approssimativamente uguali" (discuteremo tra poco il significato di questo apparentemente strano concetto), da cui potremo concludere che $y_n \approx z_n \approx \sqrt{x}$, che è il risultato che volevamo ottenere.

Siamo ora quasi pronti per tradurre questa descrizione in un algoritmo: ci manca solo di decidere quale criterio di "approssimata uguaglianza" tra coppie di numeri, $a \approx b$, intendiamo adottare. Si tratta di stabilire se la differenza tra a e b è sufficientemente prossima a zero, cioè se, per un certo valore k positivo:

–k < a–b < k

Introducendo poi la funzione valore assoluto (che "toglie il segno" a un numero, ed è indicata |x| in matematica e con la notazione funzionale, per esempio abs(x), in molti linguaggi di programmazione), si può scrivere questa condizione anche come:

abs(a–b) < k

Per la soluzione di questo problema preliminare ci rimane dunque solo da scegliere un opportuno valore k. Un criterio di scelta potrebbe essere basato sulla condizione:

- per un certo numero naturale n, $a \approx b$ se a e b hanno almeno le prime n cifre decimali uguali

Ma è un buon criterio per il nostro scopo? Vediamo un esempio:

a = 1,0000000...

b = 0,9999999...

a e b hanno addirittura tutte le cifre diverse, eppure sono molto vicini l'uno all'altro. Questo ci suggerisce che il criterio ipotizzato non è appropriato. Meglio invece:

- per un dato numero naturale n, $a \approx b$ se abs(a–b) ha la parte intera e almeno le prime n cifre decimali pari a zero

cioè se abs(a–b) è un numero della forma 0,00xxx... (per n = 2). Come deve essere scelto k in questo caso?

Affrontiamo questo problema per passi, a partire dal caso più facilmente visualizzabile in cui la condizione è:

- c è un numero non negativo la cui parte intera è zero cioè è un numero della forma 0,xxx... (dunque con n = 0) In tal caso la condizione da controllare è, evidentemente:

$c < 1$

Il caso n = 1, cioè:

- c è un numero non negativo la cui parte intera e la cui prima cifra decimale sono uguali a zero

corrisponde allora alla condizione:

$c < 0,1$

A questo punto la generalizzazione per un valore n generico è facile, ma possiamo costruire un modello di STGraph per verificare che tutto funzioni:

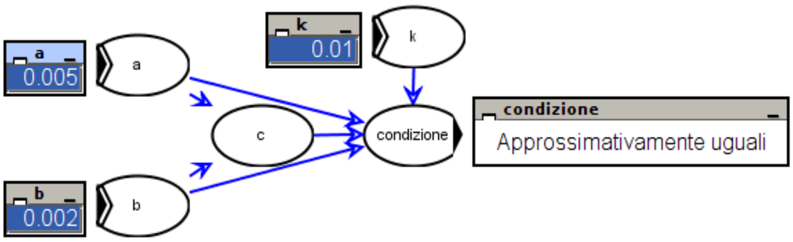
a ← numero da scegliere

b ← numero da scegliere

k ← numero da scegliere (nella forma 0,0...01)

c ← abs(a – b)

condizione ← c < k



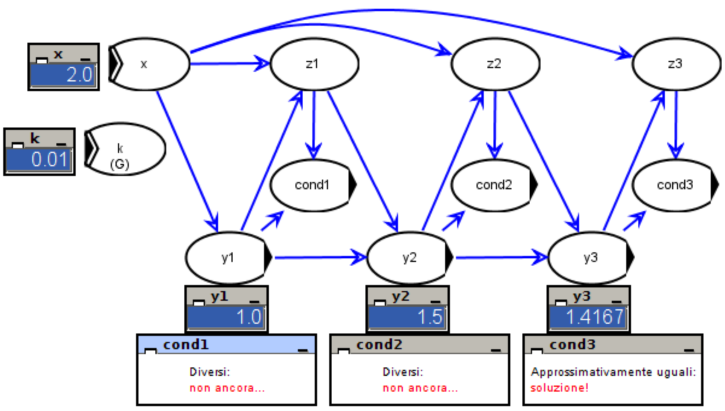
Ecco allora finalmente l'algoritmo:

1. scegli il valore x, x>0 di cui calcolare $\sqrt{x}$
2. assegna (per esempio) $y_1 \leftarrow x/2$
3. calcola $z_1 \leftarrow x/y_1$
4. verifica se $y_1 \approx z_1$: in caso positivo, il risultato è y_1 (e quindi concludi); altrimenti continua
5. calcola $y_2 \leftarrow (y_1+z_1)/2$
6. calcola $z_2 \leftarrow x/y_2$
7. verifica se $y_2 \approx z_2$: in caso positivo, il risultato è y_2 (e quindi concludi); altrimenti continua

8. calcola $y_3 \leftarrow (y_2+z_2)/2$
9. calcola $z_3 \leftarrow x/y_3$
10. verifica se $y_3 \approx z_3$: in caso positivo, il risultato è y_3 (e quindi concludi); altrimenti continua
11. ...

Perché abbiamo interrotto la descrizione all'undicesima istruzione? Ci sono due, entrambe buone, ragioni: la prima è che una successione di istruzioni di questo genere chiaramente può essere sempre proseguita, cioè, in un certo senso, potrebbe "non terminare mai", e quindi tanto vale interromperla presto; e d'altra parte – e questa è la seconda ragione – si vede chiaramente che le istruzioni 5-7 sono in pratica identiche alle istruzioni 8-10, e lo sarebbero alle istruzioni 11-13, alle istruzioni 14-16, … se solo le avessimo scritte.

Ed ecco il modello STGraph che implementa l'algoritmo fino all'istruzione 10, e dunque con tre fasi successive di approssimazione:



Sia la precedente successione di istruzioni sia questo grafo mettono bene in evidenza la possibilità di interpretare questo algoritmo secondo una struttura iterativa; come segue:

1. scegli il valore x, x>0 di cui calcolare $\sqrt{x}$
2. assegna (per esempio) $y \leftarrow x/2$
3. calcola $z \leftarrow x/y$
4. verifica se $y \approx z$: in caso positivo, il risultato è y (e quindi concludi); altrimenti continua
5. calcola $y \leftarrow (y+z)/2$
6. torna al passo 3

Scritto in questo modo, l'algoritmo è particolarmente interessante per la presenza dell'istruzione 5, in cui la stessa variabile, y, compare sia a sinistra sia a destra del simbolo di assegnazione. Una volta verificato che questo non solo non è un errore, ma anzi che è necessario se si vuole scrivere l'algoritmo in questa forma sintetica (e comunque senza "i puntini” – come nell’istruzione 11 della precedente versione – che alludono a una prosecuzione indefinita), possiamo chiederci cosa significhi un’assegnazione di questo genere.

Notiamo innanzitutto che la presenza nell’istruzione 5 di una seconda variabile, z, non ha alcuna rilevanza per il problema che intendiamo prendere in esame, che infatti non cambia se lo consideriamo in riferimento alla più semplice istruzione:

$$y \leftarrow y+1$$

Cosa significa questa istruzione? Prima di tutto, è necessario puntualizzare che non si tratta di un’equazione. Se lo fosse, cioè se stessimo cercando un valore y tale che:

$$y = y+1$$

potremmo concludere immediatamente che non esiste soluzione. Si può con ciò apprezzare perché per denotare l’assegnazione abbiamo impiegato non l’ambiguo ‘=’ ma ‘←’ , che in matematica non è usato con alcun significato analogo. In molti linguaggi di programmazione un’istruzione come la precedente viene interpretata, assai semplicemente, a partire dalle ipotesi secondo cui:

- y è una variabile numerica;
- ogni variabile numerica è univocamente associata a un’area di memoria (diciamo: “un contenitore”);
- il contenuto di tale area di memoria può essere letto e può essere sovrascritto.

L’istruzione “y = y+1” corrisponde allora a tre sotto-istruzioni:

- leggi il contenuto dell’area di memoria associata a y;
- ipotizzando che quanto è stato letto sia un numero, aggiungi 1 a tale numero;
- sovrascrivi il risultato nella stessa area di memoria.

Questa pur breve analisi mostra perché i suddetti linguaggi di programmazione si basino su una semantica cosiddetta imperativo-procedurale:

- imperativa, perché le istruzioni sono interpretate come comandi da eseguire;
- procedurale, perché le istruzioni sono da eseguire in successione, costituendo così nel loro complesso una procedura (che, tra l’altro, formalizza un algoritmo).

Non è però certo questa la semantica tipica della matematica: ciò implica che “y = y+1” non ammette un’interpretazione appropriata in matematica? Certo che no! E’ infatti sufficiente introdurre un indice che, come nell’algoritmo sopra, potrebbe rappresentare il passo dell’iterazione:

$$y_{i+1} = y_i+1$$

Questa torna ad essere un'equazione, dunque trattabile a prescindere da questioni di informatica, con la sola peculiarità di essere iterativa: il prossimo (i+1-esimo) valore della variabile y è uguale al valore attuale (i-esimo) incrementato di 1. E' poi particolarmente interessante adottare un'interpretazione fisica, e considerare l'indice i come rappresentante il tempo: mentre il tempo passa, la variabile y viene "osservata" (naturalmente in matematica diremmo: "calcolata") negli istanti successivi 1, 2, …, i, i+1, … Per enfatizzare questa interpretazione potremmo riscrivere l'equazione così:

$$y(t+1) = y(t)+1$$

Se per esempio y descrive la posizione di un corpo che si muove lungo un asse, e dunque y(t) è la posizione del corpo al tempo t, l'equazione descrive un moto rettilineo uniforme, con velocità 1 (naturalmente per rendere corretta l'analogia fisica a questo punto occorrerebbe introdurre le unità di misura). Proprio questo richiamo alla fisica ci ricorda, poi, che l'equazione può essere risolta solo se conosciamo la posizione iniziale del corpo, y(t=0), ciò che in matematica si chiama la base dell'iterazione (e così se, per esempio, y(0)=10, allora y(1)=11, y(2)=12, e così via).

In fisica una grandezza come in questo caso è y, in cui valore è calcolato iterativamente a partire da una condizione iniziale, si chiama "variabile di stato", perché descrive la condizione, lo stato appunto, in cui si trova il sistema sotto osservazione.

Possiamo dunque concludere che, in base a come il loro valore viene calcolato, le variabili si dividono in due categorie:

- le variabili il cui valore è calcolato senza la necessità di un riferimento a un indice (che potrebbe rappresentare il tempo), e che quindi possiamo immaginare siano calcolate sincronicamente con le eventuali (altre) variabili da cui dipendono. Dato che si tratta delle variabili usualmente presenti nelle equazioni, le possiamo chiamare "variabili algebriche". Nell'algoritmo precedente una variabile algebrica è z, il cui valore è ottenuto da x/y: naturalmente anche in questo caso è possibile introdurre esplicitamente un indice, $z_i = x_i/y_i$, ma ciò non fornisce alcuna informazione aggiuntiva;
- le variabili il cui valore può essere calcolato solo in riferimento a un indice, e che sono calcolate diacronicamente rispetto alle variabili da cui dipendono, nel senso che il valore è calcolato per l'indice i+1 a partire da variabili di indice i. Variabili di questa categoria possono dunque essere calcolate "in termini di se stesse", come abbiamo visto per $y_{i+1} = y_i+1$. Poiché un'equazione di questo genere è equivalente a $y_{i+1}– y_i = 1$ (un'equazione chiamata "alle differenze" proprio in contrapposizione alle usuali

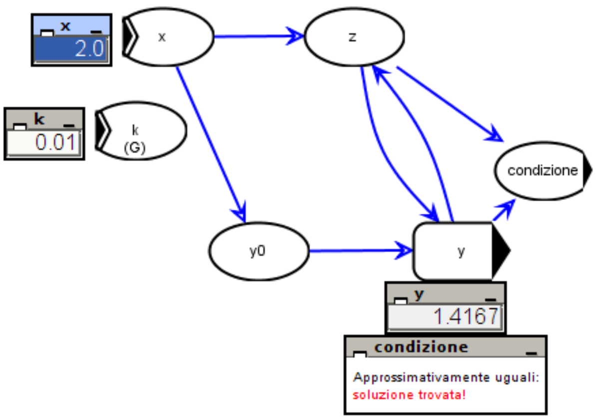
equazioni algebriche), le variabili in questione possono essere chiamate "differenziali", o anche, come abbiamo visto, "di stato". Nell'algoritmo precedente una variabile differenziale è y, il cui valore è ottenuto da (y+z)/2: per poter scrivere l'assegnazione come un'equazione è infatti necessario introdurre esplicitamente un indice, $y_{i+1} = (y_i+z_i)/2$.

Siamo ora pronti per riprendere in esame la versione iterativa dell'algoritmo, e a partire da essa costruire un modello di calcolo in STGraph. La novità è che ci è sufficiente una sola variabile y, ma che essa deve essere dunque dichiarata come di stato. In quanto tale, è necessario definire sia il valore iniziale della variabile (è ciò che in fisica si chiama "stato iniziale") sia l'equazione di ricalcolo (che può essere chiamata "equazione "di transizione di stato"):

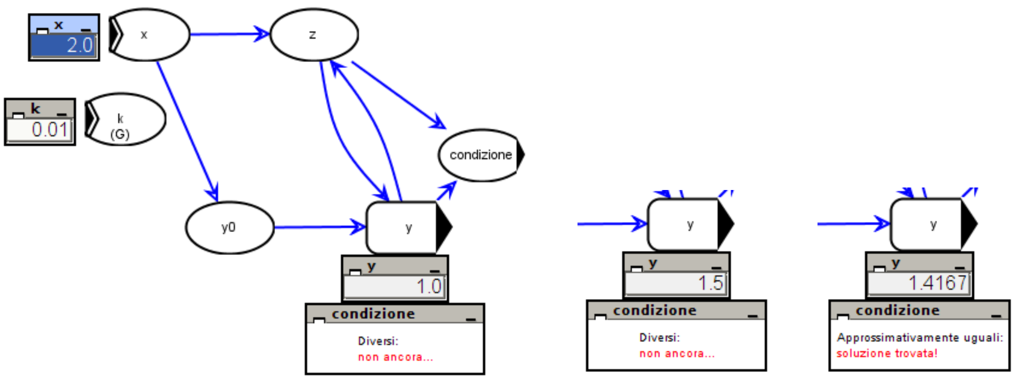
y-init ← x/2

y-trans ← (this+z)/2

(nel linguaggio di STGraph "this" rappresenta la variabile di stato che si sta definendo e ha il valore attuale della variabile stessa). Ecco il modello:



chiaramente più semplice del precedente, e in cui la nuova forma del nodo y mette in evidenza che y è una variabile di stato. Per ottenere il risultato mostrato è necessario, questa volta, eseguire una vera e propria simulazione, cioè eseguire iterativamente il calcolo simulando appunto che il tempo passi. Questa è la successione degli eventi:



tempo = 0 = 1 = 2

Il risultato è così ottenuto.

Riassunto di numeri

Note didattiche

Questa attività di laboratorio, in coerenza e insieme ad altre attività, vuole contribuire all'acquisizione da parte degli studenti delle seguenti competenze:

- individuare le strategie appropriate per la soluzione di problemi;
- progettare e costruire modelli di situazioni reali;
- analizzare dati e interpretarli sviluppando deduzioni e ragionamenti sugli stessi anche con l'ausilio di rappresentazioni grafiche, usando consapevolmente gli strumenti di calcolo e le potenzialità offerte da applicazioni specifiche di tipo informatico,

e si propone, sempre insieme ad altre attività, di raggiungere i seguenti obiettivi formativi:

- educare all'utilizzo di strumenti scientifici per leggere, interpretare e analizzare le problematiche sociali utilizzando strumenti scientifici, evitando conclusioni scaturite da “luoghi comuni”;
- sviluppare la capacità di sintesi dei dati analizzati, al fine di formulare ipotesi plausibili sull'evolversi del fenomeno osservato.

Compito

Esplorare numericamente qualche elemento basilare della statistica descrittiva e in particolare sintetizzare una stessa successione numerica secondo alcuni criteri diversi, analizzando il significato dei risultati ottenuti.

Obiettivi

Per sviluppare e/o potenziare le competenze sopra descritte si lavorerà sulle seguenti abilità:

- raccogliere, analizzare, organizzare e rappresentare un insieme di dati;

- rappresentare classi di dati mediante istogrammi e altri strumenti di rappresentazione grafica;
- leggere e interpretare tabelle e grafici in termini di corrispondenze fra elementi di due insiemi;
- riconoscere una relazione fra variabili e formalizzarla attraverso una funzione matematica;
- rappresentare il grafico di una funzione;
- elaborare e gestire semplici calcoli attraverso un foglio elettronico;
- elaborare e gestire un foglio elettronico e rappresentare in forma grafica i risultati dei calcoli eseguiti;
- accrescere e apprezzare il gusto della scoperta;
- generalizzare;
- presentare il lavoro svolto cogliendone gli aspetti più significativi.

Metodologia

- Didattica per problemi.
- Scoperta guidata.

Strumenti

- Foglio di calcolo.

Attività

Vogliamo esplorare qui numericamente qualche elemento basilare della statistica descrittiva, la disciplina che fornisce gli strumenti matematici per sintetizzare grandi quantità di numeri. Il nostro obiettivo strumentale è di arrivare a sintetizzare una stessa successione numerica secondo alcuni criteri diversi, analizzando il significato dei risultati ottenuti. Nella prima parte studiamo come può essere ottenuta o generata una successione che sia interessante da sintetizzare. Nella seconda parte introduciamo e sperimentiamo alcune tecniche di sintesi.

Prima parte: generazione

"Fare il riassunto" di un testo (un racconto, una descrizione, …) significa scrivere un nuovo testo che abbia due caratteristiche:

- sia più breve del testo iniziale,

e nonostante ciò:

- consenta di ricostruire, almeno per linee generali ma fedelmente, il contenuto del testo iniziale.

E' un'esperienza comune che quando un testo è molto lungo, e più in generale quando è disponibile una grande quantità di dati, farne il riassunto è una soluzione efficiente per riuscire a comprendere rapidamente il contenuto del testo o dell'insieme dei dati. E' possibile riassumere una successione di numeri? Con un esempio mostriamo che la risposta è positiva.

Consideriamo un'immagine digitale, come quella che potrebbe essere acquisita da una macchina fotografica digitale e quindi memorizzata in un file: si tratta di una matrice rettangolare di punti (chiamati spesso "pixel", contrazione dell'espressione inglese "picture element", cioè appunto "elemento di immagine"), che in quest'immagine sono disposti in 150 righe di 200 colonne ciascuna:

così che l'immagine stessa è costituita di $150 \times 200 = 30\,000$ punti (analogamente, le immagini ottenute da una macchina fotografica con l'elemento sensibile da 10 megapixel sono costituite di 10 milioni di punti). Ognuno di questi punti ha un colore ottenuto generalmente per combinazione di tre componenti, rosso, verde e blu (da cui il nome "modello RGB": red, green, blue), e l'intensità di ogni componente è descritta da un numero intero tra 0 (componente assente) e 255 (componente a massima intensità, cioè colore pieno):

Poiché tra 0 e 255 inclusi ci sono 256 numeri interi, cioè è possibile memorizzare 256 intensità diverse, e $256 = 2^8$, occorre proprio 1 byte per memorizzare l'intensità di una componente cromatica di un pixel. Ma allora:

$$1 \text{ byte per componente} \times 3 \text{ componenti per pixel} \times 30\,000 \text{ pixel} = 90\,000 \text{ byte}$$

necessari per memorizzare l'immagine. E infatti se memorizziamo l'immagine in un formato "raw", per esempio bpm, è proprio questa la dimensione del file che otteniamo. Se, d'altra parte, cambiamo formato e scegliamo jpg, abbiamo la possibilità di assegnare un valore a un parametro chiamato "qualità dell'immagine". Ecco il risultato, nel caso di qualità pari al 50%, 10% e 5% dell'originale rispettivamente:

Il peggioramento progressivo della qualità è, appunto, evidente! Ma confrontiamo anche le dimensioni dei file:

- originale: 90 000 byte
- qualità al 50%: 4600 byte circa
- qualità al 10%: 1600 byte circa
- qualità al 5%: 1000 byte circa

Possiamo perciò concludere che attraverso l'algoritmo che memorizza immagini in formato jpg si può "fare il riassunto" di successioni di numeri, interpretate come immagini, secondo criteri che rispettano le due condizioni riportate sopra: maggiore brevità e sufficiente ricostruibilità dell'originale. Riassumere insiemi di numeri ha dunque senso, e spesso, anzi, è molto utile!

Facciamo un secondo esempio; ecco una successione di numeri:

5, 12, 29, 56, 93, 140, 197, 264, 341, 428, 525, 632, 749, 876, 1013, 1160

che potremmo aver ottenuto misurando la posizione p di un oggetto in istanti di tempo successivi, t = 0, 1, 2, ...

Esercitando un po' delle nostre capacità di intuizione, potremmo scoprire che i valori della successione soddisfano la condizione:

$$p(t) = 5 + 2\,t + 5\,t^2$$

espressione che potremmo riconoscere come un caso particolare della legge fisica che descrive il moto uniformemente accelerato.

Ne concludiamo che le leggi della fisica sono interpretabili come esempi di strumenti per "fare il riassunto" di successioni di numeri (in questo caso dunque, a meno dell'unità di misura, di misure). Anche in questo caso, infatti, le due condizioni sono soddisfatte.

C'è comunque una differenza non secondaria tra i due esempi, a proposito della condizione di ricostruibilità dell'originale a partire dal riassunto: mentre il risultato prodotto dall'algoritmo jpg non consente di riottenere l'immagine iniziale, la sintesi generata da una legge fisica non fa perdere alcuna informazione sulla successione di partenza, che dalla legge è infatti perfettamente ricostruibile. Interpretando entrambi gli esempi come algoritmi di compressione dei dati, il caso delle leggi della fisica corrisponde a una compressione appunto "senza perdita di informazione" ("lossless"), mentre jpg funziona attraverso una compressione "con perdita di informazione" (lossy"): come vedremo, la statistica opera secondo una logica lossy: si accetta di perdere informazione, pur di ottenere risultati sintetici.

Data una successione di numeri, quale informazione si può dunque considerare accettabile perdere? Come c'è da aspettarsi, risposte diverse a questa domanda corrispondono a statistiche diverse.

Per cominciare a lavorare concretamente su questi temi, potremmo supporre che la successione di numeri su cui operare ci sia data, per esempio attraverso valori ricavati da un archivio, oppure raccolti ripetendo la misurazione di una grandezza o mediante interviste di qualche genere
(anche questi pochi e generici esempi dovrebbero bastare a convincere che la statistica non è necessariamente legata a quel, per altro piuttosto misterioso, fenomeno chiamato "casualità". Un approfondimento tra la matematica e la filosofia potrebbe svilupparsi da qui: cos'è (o eventualmente, più operativamente: come si misura) la casualità? Una possibile risposta è basata proprio sulla possibilità di compressione lossless: una successione è tanto più casuale quanto meno "contiene regolarità" e quindi quanto meno può essere compressa senza perdita di informazione).

Per lavorare a qualche esempio di sintesi di successioni numeriche, attiviamo un foglio di calcolo (la tipologia di applicazioni software indicata anche con il termine inglese "spreadsheet"), nel quale supponiamo siano già presenti i dati su cui operare. Potrebbe essere, per esempio:

	A	B	C	D	E
1	Pos	Comune	Residenti	Densità per kmq	Numero Famiglie
2	1	Roma	2546804	1981.5	1039152
3	2	Milano	1256211	6899.6	588197
4	3	Napoli	1004500	8565.7	337787
5	4	Torino	865263	6647.2	394378
6	5	Palermo	686722	4322.3	233557
7	6	Genova	610307	2505.4	274425
8	7	Bologna	371217	2637.8	17768
9	8	Firenze	356118	3477.4	159724
10	9	Bari	316532	2724.0	111319
11	10	Catania	313110	1731.0	113594
12	11	Venezia	271073	657.1	116226
13	12	Verona	253208	1225.4	109786
14	13	Messina	252026	1193.1	94142

Ma è anche interessante provare a costruire artificialmente la successione su cui poi si lavorerà, a partire da un generatore di numeri casuali: nel passato ciò avrebbe comportato lanciare monete o dadi; oggi ci si affida a funzioni software, che implementano algoritmi "pseudo-casuali". In un foglio di calcolo, la formula basilare per la generazione di numeri pseudo-casuali è:

=CASUALE()

che ogni volta che viene calcolata produce (senza un criterio evidente, e quindi, pragmaticamente, "in modo casuale") un numero decimale tra 0 e 1
(una nota a proposito di notazione: ci si potrebbe chiedere la ragione delle parentesi che seguono il nome della funzione, data l'assenza di argomenti al loro interno; tale notazione ha lo scopo di rimuovere la potenziale ambiguità circa la natura dell'entità di nome "CASUALE": la presenza delle parentesi indica che si tratta appunto di una funzione)
(una nota operativa: purtroppo, i nomi delle funzioni dei fogli di calcolo sono tradotti nelle diverse lingue, con la conseguenza che si usasse una versione inglese dell'applicazione la formula precedente dovrebbe essere scritta:

=RAND()

dall'inglese "random", "casuale" appunto. Nel seguito faremo solo riferimento ai nomi e alla notazione adottata nelle versioni italiane dei fogli di calcolo).

Copiando questa formula nelle celle da A1 ad A10 si ottiene, per esempio (ogni volta che il foglio viene ricalcolato, per esempio con il tasto F9, sono generati valori diversi):

	A
1	0.698547543
2	0.579548151
3	0.624700013
4	0.21321217
5	0.347399124
6	0.267920488
7	0.423497166
8	0.381054148
9	0.822279559
10	0.348534771
11	

Poniamoci a questo proposito un problema preliminare: usando solo la funzione CASUALE() e le quattro operazioni, modificare la definizione del nodo in modo che il numero decimale prodotto sia contenuto non nell'intervallo [0, 1] ma in un intervallo di estremi [a, b] generici.

Di fronte a un problema nuovo, e la cui soluzione non ci si presenta rapidamente, è spesso utile operare secondo il principio "divide et impera": dividi il problema in sottoproblemi, da affrontare separatamente e auspicabilmente più semplici del problema iniziale; quindi "metti insieme" i risultati così ottenuti. Nel nostro caso, potremmo prima di tutto prendere in esame il problema di produrre un numero per esempio nell'intervallo [0, 2] e dunque in generale nell'intervallo [0, p], per un certo p>0. Il modo più semplice (e in questo caso anche il più corretto) per convertire un valore in [0, 1] in un valore in [0, p] è di moltiplicarlo per p, estendendo così l'ampiezza dell'intervallo:

p*CASUALE()

Se invece volessimo produrre un valore da un intervallo ancora di ampiezza 1 ma spostato rispetto allo 0, per esempio [3, 4], e quindi in generale [q, q+1] per un certo q, allora evidentemente dovremmo sommare q:

CASUALE()+q

Componendo le due soluzioni, si ottiene perciò:

p*CASUALE()+q

Ma noi vogliamo specificare gli estremi dell'intervallo [a, b], non i parametri p e q: poiché l'estremo sinistro dell'intervallo (quando CASUALE() assume il valore 0) è dato da q e l'estremo destro (quando CASUALE() assume il valore 1) da p+q, e quindi:

[a, b] = [q, p+q]

e quindi:

q = a

p = b–a

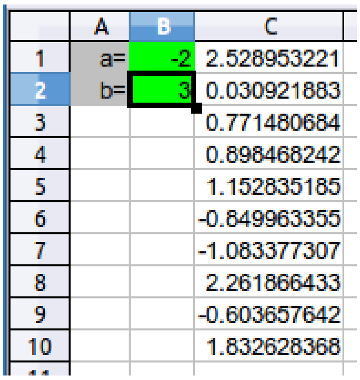
e perciò:

(b–a)*CASUALE()+a

che nel foglio sotto diventa:

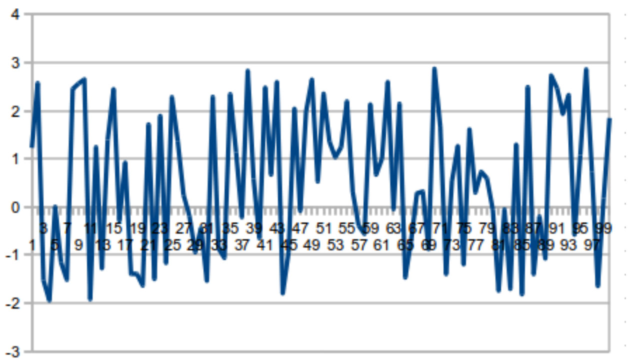
(A4–A2)*CASUALE()+A2

Su questa base, possiamo costruire quello che in statistica si chiama un campione, cioè una successione di valori che si suppongono ottenuti (generalmente senza un criterio noto, e quindi appunto “casualmente”) da un insieme dato, nel nostro caso numeri nell’intervallo [a, b].



	A	B	C
1	a=	-2	2.528953221
2	b=	3	0.030921883
3			0.771480684
4			0.898468242
5			1.152835185
6			-0.849963355
7			-1.083377307
8			2.261866433
9			-0.603657642
10			1.832628368

Se gli elementi del campione fossero sufficientemente numerosi sarebbe decisamente appropriato rappresentarli su un grafico (almeno qualche volta è vero che un’immagine vale più di mille... numeri), e dovrebbe essere chiaro che in questo caso occorre un grafico a linee, e non per esempio un istogramma, nella figura per un campione di 100 elementi:



L'indice da 1 a 100 usato in ascissa nel grafico potrebbe essere interpretato come se si riferisse al tempo: in tal caso il campione, che in statistica sarebbe chiamato "serie storica", potrebbe rappresentare i valori che una grandezza assume appunto al trascorrere del tempo. In accordo a questa interpretazione, il nostro campione non pare granché rappresentativo di fenomeni reali (fisici, economici, demografici, …), dato che, proprio per come lo abbiamo generato, ogni suo elemento è non solo casuale ma anche ottenuto indipendentemente dagli altri (in effetti qualche fenomeno fisico descrivibile in questo modo c'è, per esempio rumore nelle trasmissioni radio, la cui ampiezza ha tipicamente un andamento "completamente casuale"). E' per altro molto facile costruire serie storiche che invece siano più rappresentative e quindi più realistiche. Ecco l'idea:

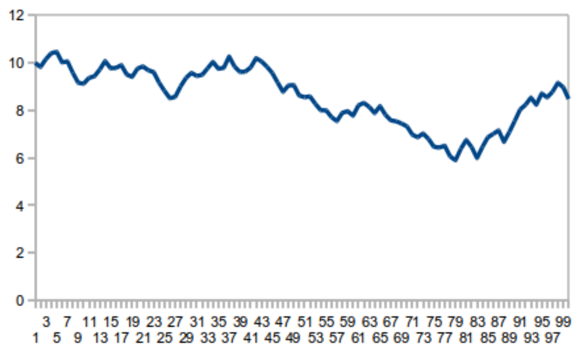
- scriviamo una costante, per esempio 10, nella prima cella, diciamo A1;
- scriviamo la formula:

  =A1+CASUALE()–0,5

  nella cella A2;

- copiamo questa formula nelle celle successive, in modo che in A3 compaia =A2+CASUALE()–0,5 e così via.

Si ottiene così quello che si chiama un "cammino casuale" (a volte si usa direttamente il termine inglese, "random walk"), per esempio:



(in questo caso si potrebbe lavorare per compiere un lavoro di traduzione non dall'italiano al matematichese, ma viceversa: data la formula e costruito il grafico, si potrebbe chiedere: cosa stiamo descrivendo? E quindi: cosa significa questa formula? E quindi, appunto: come la si può "tradurre in italiano"? Tra le varie, una possibile risposta potrebbe essere: si tratta del moto unidimensionale di un punto materiale sottoposto a forze che in ogni istante lo spostano in modo casuale, cioè di quello che si chiama "moto browniano". La posizione in ogni istante, dipende infatti dalla posizione all’istante precedente, modificata da un termine additivo casuale).

Seconda parte: sintesi

Ci poniamo finalmente il problema: indipendentemente da come sia stato ottenuto, come possiamo sintetizzare il campione?

La sintesi massima si ottiene prendendo in considerazione del campione solo il numero dei suoi elementi, calcolato mediante la funzione CONTA(x):

CONTA(C1:C10)

dato che C1:C10 è appunto l’insieme delle celle che contiene il campione.

Si può poi fare di meglio, sfruttando la possibilità che i fogli di calcolo offrono di attribuire un nome a insiemi di celle. Se chiamiamo:

campione ← C1:C10

possiamo scrivere, in modo equivalente:

CONTA(campione)

(questa forma mostra che attraverso l’attribuzione di un nome un insieme di celle può essere considerato come una variabile: una singola cella come una variabile scalare, e un insieme di più celle come un vettore. Ne possiamo ottenere, tra l’altro, un significato, semplice da comprendere e comunque in pratica corretto, del concetto di variabile: una variabile è un contenitore per valori identificato univocamente da un nome. Si noti inoltre come l’uso dei nomi delle variabili nelle formule segua la convenzione abituale circa l’impiego delle virgolette per stabilire se il riferimento sia all’entità identificata dal nome o al nome in quanto tale. Confrontiamo queste due frasi pronunciate:

<Luca sta arrivando>

<Luca è un sostantivo maschile>

La prima fa riferimento a una persona di nome Luca, mentre nella seconda si tratta proprio del nome “Luca”, e infatti la forma scritta corretta è:

<“Luca” è un sostantivo maschile>

La stessa cosa si applica (generalmente) nell’informatica, e comunque in particolare nei fogli di calcolo) (si noti che, purtroppo, i fogli di calcolo non consentono però di usare l'usuale notazione vettoriale, vettore[indice], per identificare i singoli elementi del vettore).

Quali dati sintetici più interessanti potremmo ottenere? Per esempio il valore minimo e quello massimo della successione. Lavoriamo un po' anche su questi. Prima di tutto, se il nostro sistema di calcolo non fosse già in grado di calcolare il minimo tra due numeri, come potremmo descrivere le istruzioni da eseguire?

se a < b

    allora il minimo è a

altrimenti

    il minimo è b

Corretto?

Il linguaggio dei fogli di calcolo contiene la funzione:

    SE(condizione; valore_se_vero; valore_se_falso)

e quindi la traduzione è semplice:

    SE(a<b; a; b)

D'altra parte le funzioni MAX e MIN sono già disponibili, e quindi possiamo usarle direttamente. Notiamo, poi, che vogliamo trovare il valore minimo non di due ma di tanti numeri, quelli contenuti nella successione x. Queste funzioni operano sia su coppie di argomenti, per esempio MIN(C1,C2) trova il minimo dei primi due elementi del campione, sia su interi insiemi, e possiamo perciò scrivere sia:

    MIN(C1:C10)

sia, in modo equivalente, come abbiamo visto:

    MIN(campione)

Possiamo così sintetizzare la successione mediante il suo minimo e il suo massimo, e anche, facendo "la sintesi della sintesi", con la differenza dei due, che indica l'ampiezza dell'intervallo dei valori (quello che in inglese si chiama "range"):

	A	B	C	D	E
1	a=	-2	-1.54473129	numero=	10
2	b=	3	-1.516790921	minimo=	-1.914
3			-1.351510946	massimo=	1.468
4			-1.454680572	amp intervallo=	3.3821
5			1.343729114		
6			-0.417638734		
7			-1.914161668		
8			1.467974362		
9			-0.935521427		
10			-0.35020459		

Tutti questi (numero degli elementi, valori minimo e massimo, ampiezza dell'intervallo dei valori) sono esempi, molto semplici, di statistiche, che possiamo identicamente riapplicare per esempio al campione dei valori delle densità di abitanti per chilometro quadrato delle più popolose città italiane:

G	H	I	J
Roma	1981.5	numero=	60
Milano	6899.6	minimo=	206.2
Napoli	8565.7	massimo=	8565.7
Torino	6647.2	amp intervallo=	8359.5
Palermo	4322.3		
Genova	2505.4		
Bologna	2637.8		
Firenze	3477.4		
Bari	2724.0		
Catania	1731.0		
Venezia	657.1		
Verona	1225.4		
Messina	1193.1		
Trieste	2499.5		

Ne emerge una grande differenza tra la densità minima e la massima: oltre 40 volte!

Potremmo allora chiederci come sono distribuiti i valori di densità delle 60 città considerate nell'intervallo dei valori di densità possibili, se cioè per esempio ci sono molte città ad alta densità, o viceversa, o ancora se la maggior parte delle città ha una densità intermedia. Lo strumento per rispondere a questa domanda è appunto una distribuzione di frequenze: occorre:

- dividere l'insieme dei valori possibili, chiamato "insieme supporto", in opportuni sottoinsiemi, dette "categorie";
- contare quindi elementi del campione sono inclusi in ogni categoria.

Una distribuzione così ottenuta può essere infine visualizzata mediante un istogramma.

Dato che questa operazione è piuttosto complessa da realizzare con un foglio elettronico, cominciamo ad affrontare una sua versione semplificata, nel caso in cui le categorie coincidano con gli elementi dell'insieme supporto. A questo proposito, torniamo all'esempio precedente del campione generato mediante la funzione CASUALE(), cercando di modificarlo in modo che soddisfi questa condizione. La soluzione più semplice consiste nel convertire i numeri decimali del campione in numeri interi, così che le categorie siano poi (in riferimento all'esempio presentato nella figura) –2, –1, …

A questo scopo i fogli di calcolo mettono a disposizione la funzione INT(x), che restituisce il numero a cui è applicata arrotondato all'intero inferiore (e quindi non il numero senza la sua parte decimale: si verifichi la differenza nel caso dei numeri negativi!)

Si potrebbe inserire qui una variante: come si fa, usando solo la funzione INT(x) e le quattro operazioni, ad arrotondare all'intero più vicino invece che all'intero inferiore? La logica della risposta è: se la parte decimale di x è nell'intervallo tra 0 e 0,5, INT(x) opera già correttamente; altrimenti il valore da restituire è INT(x+1). Per ottenere questo comportamento, occorre calcolare la parte decimale di x. Chi trova come si fa? Naturalmente:

x–INT(x)

Allora, usando la funzione SE presentata sopra, la soluzione è:

SE(x–INT(x)<=0,5;INT(x);INT(x+1))

(in assenza del simbolo ≤ sulla tastiera dei calcolatori, si usa generalmente <= con lo stesso significato). Una volta compresa questa logica, si può individuare una soluzione più diretta:

INT(0,5+x)

(incidentalmente, i fogli di calcolo mettono a disposizione la funzione ARROTONDA proprio con questo scopo:

ARROTONDA(x;0)

in cui il secondo argomento indica il numero di decimali da mantenere).

Più complesso è il problema generale: arrotondare un numero x non all'intero più vicino, ma al multiplo più vicino di un numero y dato. Con un po' di riflessione si può trovare la soluzione, che appunto non è altro che una generalizzazione della precedente:

INT((y/2+x)/y)*y

Come si vede, si tratta di una soluzione già piuttosto sofisticata (anche in questo caso esiste una funzione predefinita: ARROTONDA.MULTIPLO. Questo consente di porre un problema didattico: in situazioni come queste, in cui è possibile sia “lavorare con mattoni elementari” per costruire una soluzione sia “sfruttare prefabbricati”, quale opzione adottare? Si presenta evidentemente una contrapposizione di obiettivi: il lavoro “con mattoni” consente di comprendere meglio la logica di ciò che si sta facendo, ma allunga i tempi necessari per raggiungere l’obiettivo; al contrario il lavoro “con prefabbricati”).

Predisponiamo dunque il foglio in modo da includere il campione a valori arrotondati (chiamiamolo “campione2”: nella figura è in colonna D) e lo schema della distribuzione:

	A	B	C	D	E	F	G
1	a=	-2	-1.7109191	-2		categorie	frequenze
2	b=	3	-0.467476654	-1		-2	
3			-0.357908682	-1		-1	
4			0.722046946	0		0	
5			-1.025095403	-2		1	
6			2.13287438	2		2	
7			1.26861484	1			
8			1.410491161	1			
9			-0.845241833	-1			
10			2.701147075	2			

Si tratta ora, per ogni categoria, di contare quanti elementi del campione modificato hanno valore uguale alla categoria stessa. Al proposito i fogli di calcolo mettono a disposizione la funzione:

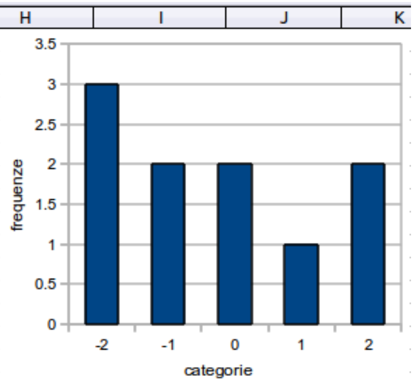
CONTA.SE(x;criterio)

che applicata per esempio nella cella G2 della figura diventa:

CONTA.SE(campione2;F2)

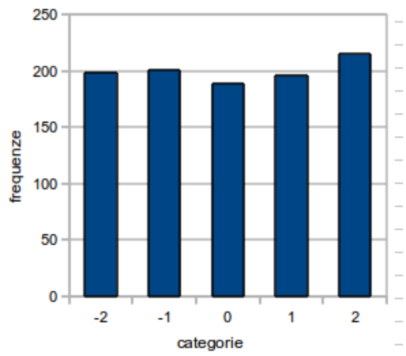
Ecco il risultato, anche nella forma di istogramma:

	A	B	C	D	E	F	G
1	a=	-2	-1.116605469	-2		categorie	frequenze
2	b=	3	2.368752888	2		-2	3
3			-0.770052276	-1		-1	2
4			-1.777487742	-2		0	2
5			-0.850856633	-1		1	1
6			0.988227324	0		2	2
7			-1.653422632	-2			
8			1.348797171	1			
9			2.501901169	2			
10			0.120251835	0			



Un po' più interessante è la stessa distribuzione se calcolata su un campione più numeroso, per esempio di 1000 elementi:

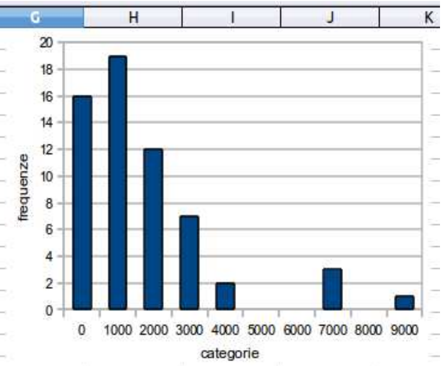
	A	B	C	D	E	F	G
1	a=	-2	0.108676282	0		categorie	frequenze
2	b=	3	0.546367866	0		-2	199
3			-1.329782936	-2		-1	201
4			-1.583464414	-2		0	189
5			-1.927138872	-2		1	196
6			0.467432816	0		2	215
7			1.314002852	1			
8			0.501114332	0			
9			-0.80395937	-1			
10			1.404934695	1			
11			-0.319791933	-1			
12			-0.679725798	-1			
13			0.937478833	0			
14			2.987745215	2			
15			-0.064756815	-1			
16			[illegible]	[illegible]			



Si vede che la distribuzione generata dalla funzione CASUALE() è approssimativamente uniforme, cioè le frequenze delle varie categorie sono approssimativamente le stesse.

La stessa logica si può applicare al campione delle densità di abitanti per chilometro quadrato delle città italiane, a meno di arrotondare appropriatamente i valori del campione (abbiamo discusso sopra come fare), per esempio al migliaio (con la conseguenza quindi che i valori minori di 500 sono inclusi nella categoria 0). Ecco il risultato:

	A	B	C	D	E	F
1	Roma	1981.5	2000		categorie	frequenze
2	Milano	6899.6	7000		0	16
3	Napoli	8565.7	9000		1000	19
4	Torino	6647.2	7000		2000	12
5	Palermo	4322.3	4000		3000	7
6	Genova	2505.4	3000		4000	2
7	Bologna	2637.8	3000		5000	0
8	Firenze	3477.4	3000		6000	0
9	Bari	2724.0	3000		7000	3
10	Catania	1731.0	2000		8000	0
11	Venezia	657.1	1000		9000	1
12	Verona	1225.4	1000			
13	Messina	1193.1	1000			
14	Trieste	2499.5	2000			
15	Padova	2206.5	2000			
16	Taranto	928.9	1000			



I passi successivi potrebbero essere di lavorare a proposito delle statistiche di posizione, mediana e media, sulle statistiche di dispersione, percentili e varianza / deviazione standard, e quindi finalmente sulle statistiche di correlazione su campioni bi- o multi-variati.

Appendici

Queste Appendici introducono ad alcuni aspetti, sia operativi sia concettuali, di STGraph, il sistema software impiegato nelle prime quattro attività di didattica laboratoriale presentate sopra.

STGraph è un sistema open source e la sua licenza consente il suo utilizzo libero per attività no profit come quelle che tipicamente una scuola potrebbe farne.

Una breve introduzione operativa a STGraph

STGraph è un sistema software per creare, modificare ed eseguire modelli di sistemi dinamici descritti secondo l'approccio agli stati della Teoria dei Sistemi. Un modello in STGraph è un grafo, i cui nodi e frecce rappresentano rispettivamente variabili e relazioni di dipendenza funzionale tra variabili.

STGraph consente di costruire i grafi con le tecniche usuali dei programmi a interfaccia grafica interattiva. Come è abituale, lo stesso risultato può essere ottenuto generalmente con tecniche diverse (selezione di una voce di menu, selezione di un'icona nella barra degli strumenti, menu contestuale aperto con click del tasto di destra del mouse, scorciatoia di tastiera). In questa Introduzione, si fa riferimento alle voci di menu in versione italiana.

Prima di cominciare

STGraph è un'applicazione scritta in Java, che può essere eseguita su calcolatori con sistema operativo MSWindows, Apple MacOS o Linux. E' solo necessario che sia installata una Java virtual machine (JRE o JVM), versione 1.5 o successiva (scaricabile da http://java.com/download).

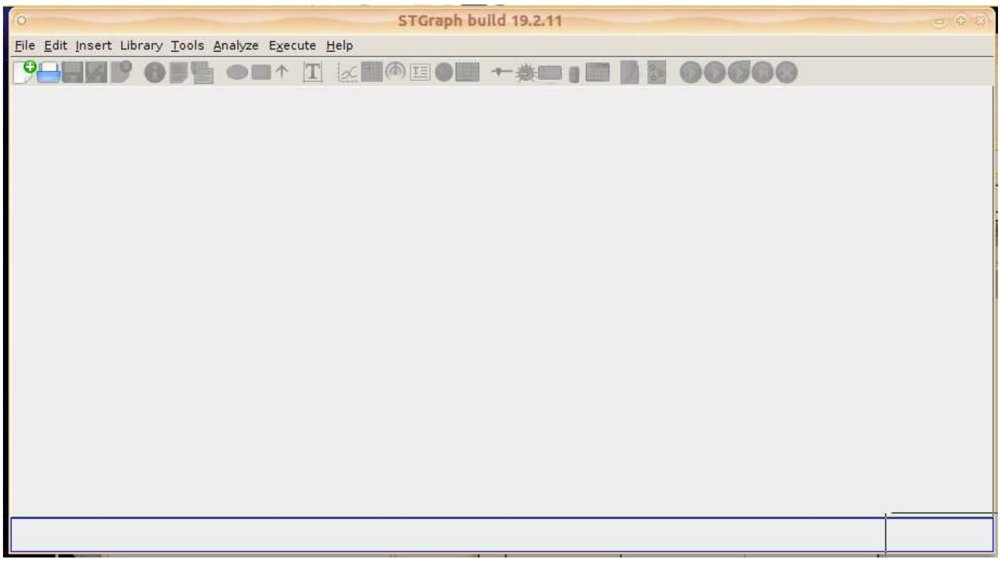
La versione più recente di STGraph e della sua documentazione è scaricabile liberamente all'indirizzo:

http://cetic.liuc.it/stgraph

Una volta che il file zip scaricato è stato decompresso, si attiva l'applicazione eseguendo stgraph.bat o stgraph.exe, se in ambiente MSWindows, o stgraph.sh, se in ambiente Apple MacOS o Linux.

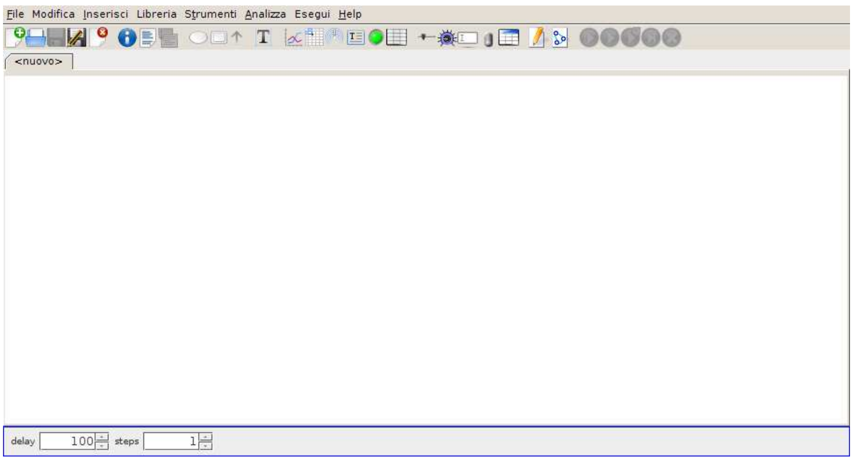
Per cominciare

All'apertura di STGraph viene visualizzata una finestra vuota:



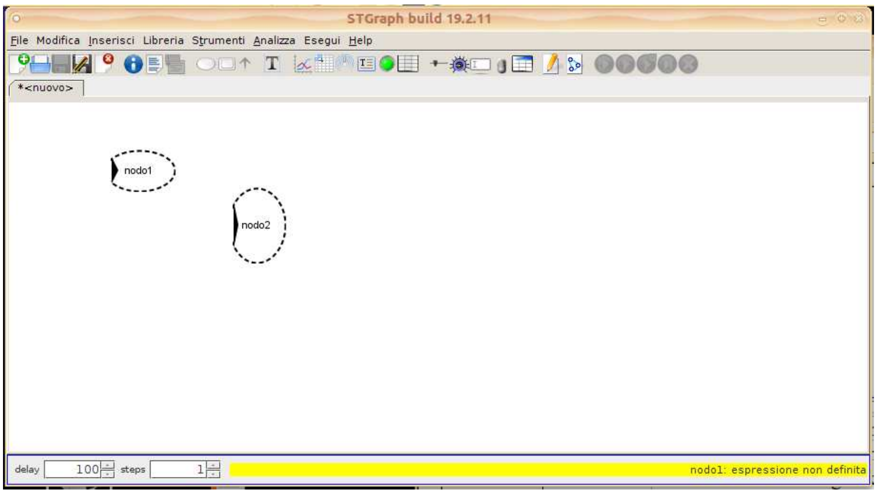
E' possibile modificare la lingua dei testi dell'interfaccia utente, se in inglese o in italiano, dal menu `[Tools | Preferences...]` (in italiano `[Strumenti | Configurazione...]`), riavviando quindi l'applicazione.

Per creare un nuovo modello, selezioniamo `[File | Crea Nuovo Modello]`. Questo è il risultato:



Un modello vuoto, che si presenta come un foglio vuoto, è stato così creato.

Introduciamo due nodi con `[Inserisci | Inserisci Nodo Variabile]`, che si possono spostare e ridimensionare nel foglio con le usuali tecniche.

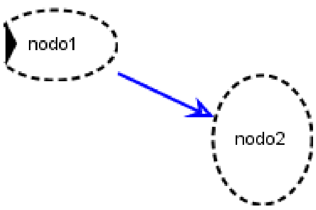


Si noti:

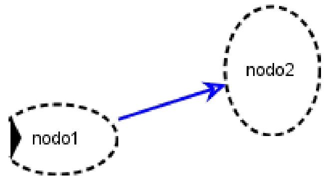
- il contorno tratteggiato dei nodi: significa che non sono stati ancora completamente definiti, oppure che le equazioni con cui sono stati definiti contengono degli errori;
- la barra di stato, in giallo, in fondo a destra nella finestra: indica il nome del primo nodo la cui definizione è errata, insieme con una breve descrizione dell'errore; facendo click sulla barra viene selezionato ed evidenziato il nodo in questione;
- i pulsanti all'estrema destra della barra degli strumenti, disattivi (corrispondenti alle voci del menu `Esegui`): si tratta dei pulsanti che consentono di controllare l'esecuzione della simulazione sul modello, che non è per ora possibile appunto a causa della definizione non ancora completa.

Si noti inoltre che nel profilo dei due nodi è presente una freccia entrante: è l'indicazione che al momento sono nodi di input, cioè corrispondenti a variabili esogene al modello.

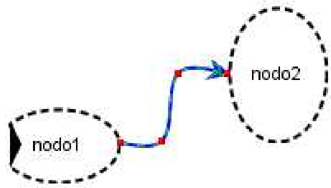
Portando il cursore del mouse sul centro del nodo `nodo1`, con un'azione di trascinamento fino a `nodo2`, creiamo una freccia:



che rimane connessa ai nodi se questi vengono spostati:



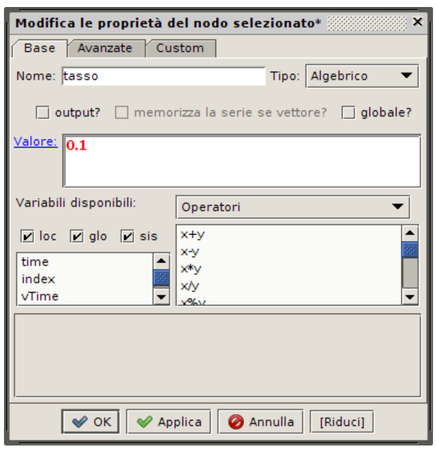
e la cui forma può essere modificata aggiungendo (ed eventualmente poi eliminando) punti di controllo facendo click con il tasto `maiusc` premuto:



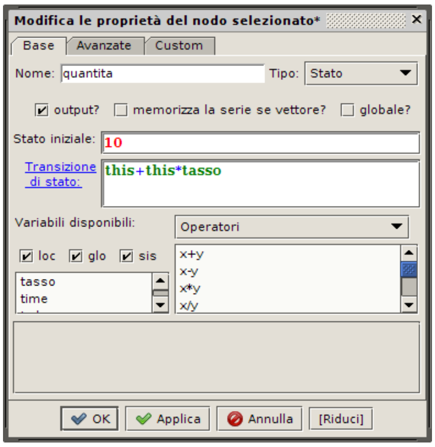
In ogni istante, il programma controlla che il grafo sia corretto, e in particolare vieta l'introduzione di frecce parallele con lo stesso verso e di frecce da un nodo a se stesso. Il programma stabilisce anche in modo automatico la forma dei nodi: come vedremo, l'ellisse corrisponde a variabili algebriche.

Facendo un doppio click su ogni nodo, si apre una finestra che consente di assegnare o modificare le proprietà del nodo stesso:

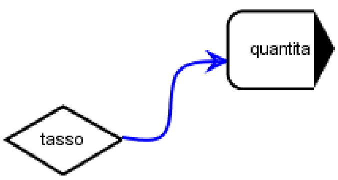
- cambiamo il nome del primo nodo, da `nodo1` a `tasso` (si noti che STGraph distingue maiuscole da minuscole, e quindi per esempio `tasso` è diverso da `Tasso`) e assegniamo `0.1` come valore (si noti che anche con interfaccia in italiano STGraph mantiene la notazione numerica anglosassone, e usa quindi il punto come separatore decimale); così si presenta dunque la finestra delle proprietà per il primo nodo:



- cambiamo ora il nome del secondo nodo, da `nodo2` a `quantita` (si noti che i nomi dei nodi, e quindi delle variabili associate, seguono le usuali convenzioni, e quindi devono cominciare con un carattere alfabetico e non possono contenere caratteri speciali, spazio, lettere accentate, …), il tipo del nodo da "algebrico" a "stato", e assegniamo `10` come stato iniziale e `this+this*tasso` come transizione di stato (nel linguaggio di STGraph, `this` significa "il mio valore attuale" per un nodo di stato, e quindi questa espressione calcola il nuovo stato come l'attuale sommato a se stesso moltiplicato per il valore di `tasso`); attiviamo anche l'opzione "output?"; così si presenta dunque la finestra delle proprietà per il secondo nodo:



e questo è il grafo che si ottiene:

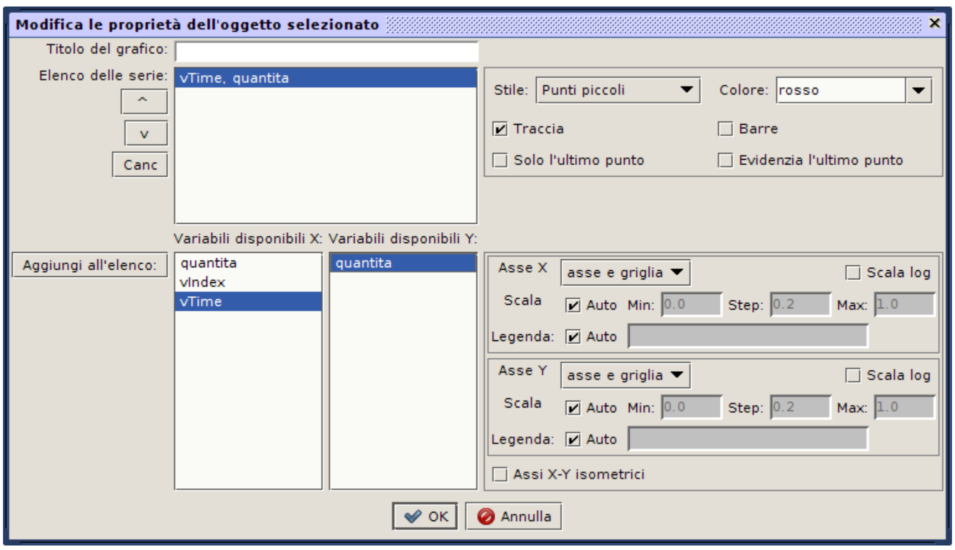


Il profilo del nodi è ora continuo, a indicare che essi sono stati definiti correttamente. Inoltre, la forma dei nodi è cambiata: `tasso` è un rombo, a indicare che è una costante, e `quantita` è un rettangolo, a indicare che un nodo di stato. La freccia uscente nel profilo del nodo `quantita` evidenzia che esso è stato dichiarato di output, e quindi sarà osservabile da oggetti esterni.

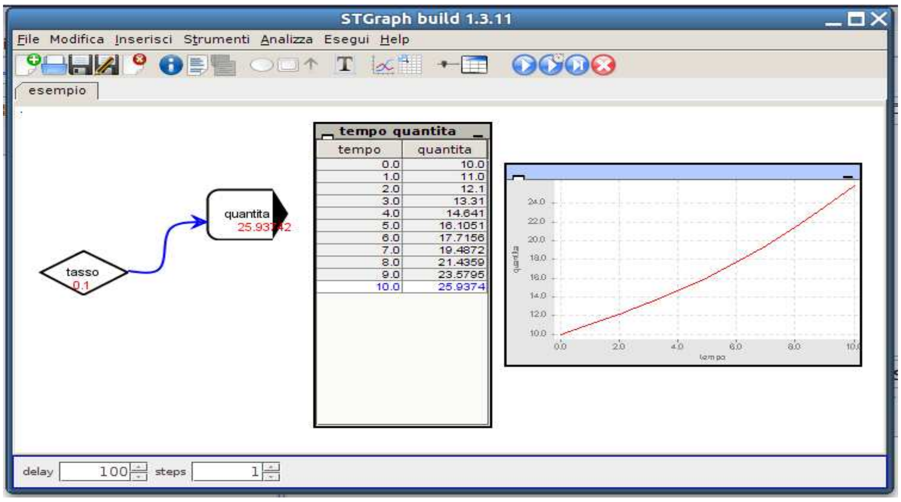
Proprio allo scopo di rendere osservabile l'andamento dei valori di questa variabile, introduciamo ora due di questi oggetti, una tabella, `[Inserisci | Inserisci Tabella come Oggetto di Output]`, e un grafico, `[Inserisci | Inserisci Grafico` `come Oggetto di Output]`, e modifichiamo le loro proprietà facendo doppio click sulle loro immagini, per ora vuote:

- nel caso della tabella, scegliamo per esempio di visualizzare le serie di valori di `vTime` (la variabile di sistema che contiene i vettori del tempo simulato) e di `quantita` (si noti che il riferimento a `quantita` compare perché per tale nodo avevamo attivato l'opzione "output?"), lasciando pure tutte le opzioni di visualizzazione con i loro valori di default; così si presenta dunque la finestra delle proprietà per la tabella:

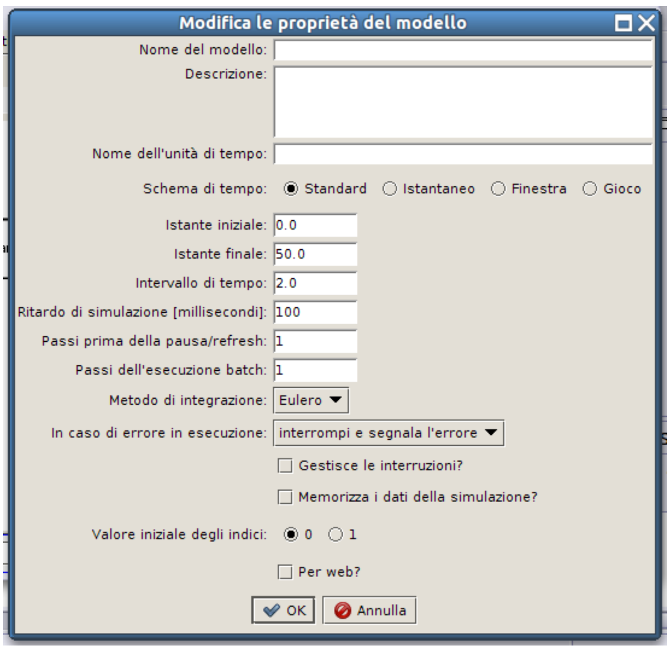
- nel caso del grafico, scegliamo di visualizzare la serie di valori di `vTime` in ascissa e di `quantita` in ordinata, anche in questo caso lasciando i valori di default per le opzioni di visualizzazione; così si presenta dunque la finestra delle proprietà per il grafico:



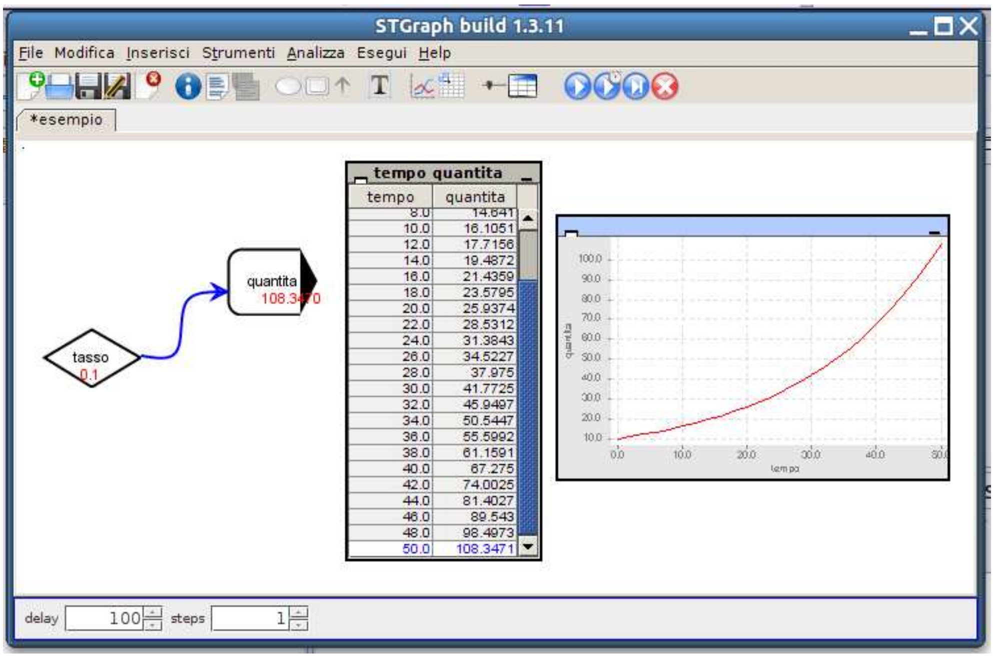
Il modello può essere ora simulato, scegliendo `[Esegui | Esegui Simulazione]`, o in modo interattivo con le voci corrispondenti dello stesso menu. Questo è il risultato:



La prima colonna della tabella mostra che la simulazione è stata calcolata sui valori di default del tempo, da `0` a `10` con passo `1`. Possiamo modificare tali valori dalla finestra delle proprietà globali del modello, `[Modifica | Modifica la Definizione del Modello]` (o anche facendo doppio click sullo sfondo della pagina), per esempio per assegnare `50` come tempo finale e `2` come passo. Così si presenta la finestra delle proprietà per il modello:



e questo è il risultato della nuova simulazione:



Una breve introduzione concettuale a STGraph

STGraph è un sistema software per creare, modificare ed eseguire modelli di sistemi dinamici. E’ stato progettato in modo che i vincoli da soddisfare nella costruzione dei modelli corrispondano fedelmente al cosiddetto approccio agli stati della Teoria dei Sistemi.

Un modello in STGraph è descritto come un insieme di variabili, il valore di ognuna delle quali può dipendere dal tempo e dal valore di ogni altra variabile del modello stesso. Le variabili condividono una stessa base dei tempi, caratterizzata da:

- istante iniziale della simulazione (variabile di sistema `time0`);
- istante finale della simulazione (variabile di sistema `time1`);
- passo temporale della simulazione (variabile di sistema `timeD`).

Le variabili sono dunque calcolate in modo sincrono in funzione della variabile di sistema `time`, che assume i valori `time0`, `time0+timeD`, `time0+2*timeD`, ...

Il valore di ogni variabile viene calcolato secondo logiche diverse, in funzione del tipo della variabile stessa.

- Una variabile algebrica ha un valore determinato attraverso la sua funzione di output, che può consistere in un valore costante (come 1 e 2+2) oppure in un’espressione dipendente esplicitamente dal tempo (come time^2) e/o da altre variabili (come x+y se x e y sono variabili definite nel modello: l’eventuale dipendenza dal tempo di tali variabili rimane implicita).

  La funzione di output è calcolata sincronicamente: dal punto di vista del modello, il valore della variabile è calcolato nello stesso istante di tempo simulato in cui sono calcolati i valori delle variabili da cui dipende; la funzione di output è dunque in effetti un'equazione algebrica.

Dal punto di vista della Teoria dei Sistemi, una variabile algebrica può essere dunque intesa come il più semplice sistema combinatorio, descritto da una funzione input → output:

η: U → Y $\quad\quad$ $y(t) = \eta(u(t))$

- Una variabile di stato ha un valore determinato attraverso le sue funzioni di stato iniziale e di transizione di stato, secondo la seguente logica:
  - nell'istante iniziale della simulazione, in cui il valore di `time` è `time0`, vengono calcolate sia la funzione di stato iniziale, e il suo valore è assegnato come stato iniziale, sia la funzione di transizione di stato, e il suo valore è assegnato come stato successivo (cioè per l'istante in cui il valore di `time` è `time0+timeD`);
  - in ogni istante successivo, lo stato attuale è determinato dal valore della funzione di transizione di stato calcolato all'istante precedente, e la funzione di transizione di stato è ricalcolata per determinare lo stato successivo, possibilmente anche in funzione dello stato attuale, indicato con la variabile di sistema `this`;
  - in ogni istante, il valore della variabile coincide con lo stato della variabile stessa in quell'istante.

La funzione di transizione di stato è calcolata diacronicamente: dal punto di vista del modello, lo stato è calcolato per l'istante di tempo simulato successivo a quelli in cui sono calcolati i valori delle variabili da cui dipende; la funzione di transizione di stato è dunque in effetti un'equazione alle differenze.

Dal punto di vista della Teoria dei Sistemi, una variabile di stato può essere dunque intesa come il più semplice sistema sequenziale, descritto da una funzione input × stato → stato:

φ: U × X → X $\quad\quad$ $x(t) = \varphi(u(t), x(t))$

η: X → Y $\quad\quad$ $y(t) = x(t)$

- Una variabile di stato con output ha un comportamento analogo a una variabile di stato, con la differenza che il suo valore non coincide con lo stato ma è determinato da una funzione di output.

Dal punto di vista della Teoria dei Sistemi, una variabile di stato con output è dunque descritta da una coppia di funzioni input $\times$ stato $\rightarrow$ stato e input $\times$ stato $\rightarrow$ output:

φ: U $\times$ X $\rightarrow$ X $x(t) = \varphi(u(t), x(t))$

η: U $\times$ X $\rightarrow$ Y $y(t) = \eta(u(t), x(t))$

Qualche semplice esempio può essere utile per mettere in evidenza le caratteristiche del comportamento delle variabili in STGraph (indichiamo con `x-init`, `x-trans`, e `x-out` rispettivamente le funzioni di stato iniziale, transizione di stato e output della variabile `x`).

x algebrica:

x-out ← sin(y)   x calcola il seno della variabile y

x algebrica:

x-out ← exp(time)   x calcola l'esponenziale della variabile di sistema time

x di stato:

x-init ← 0
x-trans ← this+1   x calcola la successione 0, 1, 2, ...

x di stato:

x-init ← u
x-trans ← this   x mantiene il valore assegnato all'istante iniziale

x di stato:

x-init ← 0
x-trans ← u   x assume il valore ricevuto in input all'istante precedente

x di stato con output:

x-init ← ...
x-trans ← u
x-out ← (u-this)/timeD   x calcola la derivata discreta della successione u

x di stato:

x-init ← ...
x-trans ← this+u*timeD   x calcola l'integrale discreto della successione u

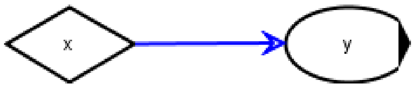
Una breve introduzione a STEL (STGraph Expression Language)

Premessa

STEL (STGraph Expression Language) è il *domain-specific language* con cui è definita la componente quantitativa dei modelli di STGraph.

Modelli come grafi; variabili come nodi; frecce come relazioni di dipendenza funzionale

Un modello in STGraph è descritto qualitativamente come un grafo orientato, in cui i nodi sono variabili (o, come caso particolare, costanti) e le cui frecce descrivono relazioni di dipendenza funzionale.



Una freccia dal nodo x al nodo y specifica che y dipende funzionalmente da x.

La forma analitica di tale dipendenza è scritta come un'espressione in STEL.

Le finestre di configurazione dei nodi della Figura precedente.
Il contenuto corrisponde alle assegnazioni x←1; y←x+1.

Il fatto che STGraph ammetta tre tipi di nodi (algebrici, di stato, di stato con output) non ha conseguenze, e quindi si tratterà qui genericamente di *espressioni in STEL*, che nei casi specifici potrebbero essere espressioni corrispondenti a una funzione di output, a uno stato iniziale, o a una funzione di transizione di stato locale.

Le caratteristiche principali di STEL in sintesi

Linguaggio funzionale

STEL è un linguaggio funzionale puro (dunque non contiene "istruzioni" né "comandi", ma appunto solo funzioni), ed è basato sull'ipotesi che le uniche relazioni ammesse tra variabili sono quelle descritte mediante le frecce del grafo (nell'esempio delle Figure precedenti, la funzione che definisce y include un riferimento a x, ma il viceversa non potrebbe essere data l'assenza di una freccia da y a x).

Assenza di istruzioni / funzioni di assegnazione di valore a variabili

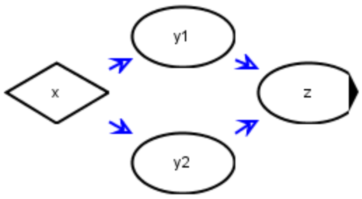
Per rendere l'ambiente di esecuzione perfettamente immune da effetti collaterali (*side effect*), STEL non include alcuna funzione per l'assegnazione di valore a variabili (ma vedi sotto a proposito di sottoespressioni): il valore di una variabile è sempre e solo assegnato mediante l'espressione associata al nodo corrispondente alla variabile stessa. In questo modo, ogni nodo del grafo corrisponde a un micro-ambiente computazionale, aperto in input solo attraverso le frecce in ingresso al nodo e in output solo attraverso il valore prodotto dalla valutazione dell'espressione.

Scope delle variabili

Ciò definisce lo scope di default delle variabili, ognuna delle quali può essere comunque resa visibile globalmente al grafo (STGraph gestisce anche modelli strutturati in sottomodelli, corrispondenti a grafi in cui uno o più nodi sono associati non a variabili ma ad altri grafi: lo scope rimane sempre a livello di sottomodello, e ogni supermodello interagisce con i suoi sottomodelli solo attraverso le variabili di input e di output dei sottomodelli stessi – tutte e sole le variabili senza frecce entranti sono di input; ogni variabile può essere dichiarata di output).

Ordine di valutazione delle variabili

L'ordine di valutazione delle espressioni nei diversi nodi è inferito da STGraph in base alla topologia del grafo: questo fa sì che, in linea di principio, sia ammessa la valutazione parallela di nodi non reciprocamente dipendenti (in presenza di variabili di stato, un grafo può contenere anche loop).



Grafo in cui le variabili `y1` e `y2` potrebbero essere valutate in parallelo.

Variabili di un unico tipo

Dato il suo dominio applicativo e per ragioni di semplicità, le variabili in STEL hanno tutte un unico e implicito (dunque non dichiarato) tipo di base, numero, che in base al contesto è interpretato come numero intero, numero decimale, o costante logica (i numeri strettamente positivi sono associati alla costante 'vero').

Variabili come array

Le variabili in STEL sono tutte di un unico tipo strutturato, array, che include come casi particolari scalari, vettori, matrici, e che ammette anche strutture tensoriali di dimensione superiore. Gli array sono dinamici, e non devono essere predefiniti.

Funzioni predefinite e funzioni definite dall'utente

STEL è dotato di un insieme di funzioni predefinite (scritte in Java come STGraph) (alcune di queste scritte nell'usuale notazione operatoriale infissa), e di una funzione per la definizione in STEL di nuove funzioni, anche in forma ricorsiva, sia in nodi di un grafo (in tal caso lo scope è il grafo stesso) sia in file XML esterni (in tal caso le funzioni sono utilizzabili da qualsiasi grafo).

Polimorfismo per funzioni a un argomento

Le funzioni predefinite sono generalmente polimorfe rispetto alla dimensione di array dei loro argomenti. Nel caso di funzioni a un argomento, ciò corrisponde, per esempio, al fatto che la funzione `sqrt(x)` ha il seguente comportamento (in questi esempi '`=`' è il simbolo metalinguistico per 'ha valore'):

se `x=4` allora `sqrt(x)=2`

se `x=[4,9]` allora `sqrt(x)=[2,3]`

se `x=[[4,9],[16,25]]` allora `sqrt(x)=[[2,3],[4,5]]`

e così via.

Polimorfismo per funzioni a due argomenti

Nel caso di funzioni a due argomenti, il polimorfismo è garantito in tutti i casi in cui la valutazione sia ammessa. Per esempio:

se `x=1` e `y=2` allora `x+y=3`

se `x=1` e `y=[1:3]` allora `x+y=[2,3,4]`

se `x=[[1,2],[3,4]]` e `y=[5,6]` allora `x+y=[[6,7],[9,10]]`

dove quest'ultimo caso è da intendere come:

$$\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} + \begin{bmatrix} 5 \\ 6 \end{bmatrix} = \begin{bmatrix} 6 & 7 \\ 9 & 10 \end{bmatrix}$$

cosa che mostra che i vettori sono interpretati come vettori colonna, che gli array a più dimensioni sono linearizzati in modo da mantenere come ultima dimensione quella più innestata, e che, quando si applica, è sull'ultima dimensione degli array che le funzioni operano.

Non includendo istruzioni, STEL non ha strutture di controllo tradizionali.

If

E' invece dotato di una funzione condizionale generalizzata, $\texttt{if}(c_1,v_1,\ldots,c_n,v_n,v_{n+1})$, a numero dispari ≥3 di argomenti e tale che gli argomenti in posizione dispari salvo l'ultimo sono condizioni c_i e gli altri sono valori v_j: la prima condizione vera seleziona il valore immediatamente successivo, e se tutte le condizioni sono false è selezionato l'ultimo valore. Per esempio:

se `x=if(1,2,3)` allora `x=2`

dato che tale funzione corrisponde alla struttura di controllo in Java:

```
if(1) {
    x=2;
} else {
    x=3;
}
```

ma anche:

se `x=if(0,2,0,3,4)` allora `x=4`

che infatti corrisponde a:

```
if(0) {
    x=2;
} else if(0){
    x=3;
} else {
    x=4;
}
```

e anche, in modo polimorfo:

```
if([1,0],2,3)=[2,3]
```

Loop temporali

Nei modelli di STGraph non è generalmente necessario introdurre in modo esplicito dei loop. Ciò è principalmente dovuto al fatto che nel corso della simulazione il tempo viene incrementato a passi discreti e quindi crea le condizioni perché, in linea di principio, l'espressione di ogni variabile sia valutata in loop (in particolare nella funzione di transizione di stato locale di un nodo di stato, per esempio un'espressione come:

```
this+1
```

– dove `this` denota lo stato attuale – incrementa di 1 lo stato a ogni istante della simulazione).

Per il caso in cui sia necessario operare in loop sincronicamente sugli elementi di un array `x` (dunque con un loop "spaziale" invece che temporale), se la funzione /operatore `f` da applicare è a due argomenti (come si è visto, funzioni a un argomento sono applicabili direttamente agli array) STEL dispone di meta-operatori che iterano la valutazione della funzione sull'array secondo logiche diverse:

- l'espressione `f/x`, dove '`/`' denota il meta-operatore *reduction*, corrisponde, nel caso `x` sia un vettore, allo scalare `f(f(x[0],x[1]),x[2])...`; per esempio:

  ```
  +/[1:4]=((1+2)+3)+4
  ```

- l'espressione `f\x`, dove '`\`' denota il meta-operatore *scan*, corrisponde, nel caso `x` sia un vettore, al vettore `[x[0],f(x[0],x[1]),f(f(x[0],x[1]),x[2]),...]`; per esempio:

  ```
  +\[1:3]=[1,1+2,(1+2)+3]
  ```

- \- l'espressione `f|x`, dove '`|`' denota il meta-operatore *pairscan*, corrisponde, nel caso `x` sia un vettore, al vettore `[f(x[0],x[1]),f(x[1],x[2]),f(x[2],x[3]),...]`; per esempio:

  ```
  +|[1:4]=[1+2,2+3,3+4]
  ```

Se applicati a matrici o array di dimensione superiore, questi meta-operatori valutano la funzione `f` lungo l'ultima dimensione. Per esempio:

```
+/[[1,2,3],[4,5,6],[7,8,9]]=[1+2+3,4+5+6,7+8+9]
```

Loop spaziali

Loop spaziali su array `x` possono essere computati più in generale mediante la meta-funzione `iter(x,e,z)`, in cui `e` è l'espressione da valutare iterativamente sugli elementi (dell'ultima dimensione) di `x` e `z` è lo "zero" di tale espressione. La valutazione è ripetuta per un numero di volte pari al numero di elementi (dell'ultima dimensione) di `x`. L'espressione `e` può contenere le variabili di sistema:

- `$i`, che a ogni iterazione ha come valore l'indice dell'iterazione;
- `$0`, che inizialmente ha valore `z` e a ogni iterazione ha come valore il risultato ottenuto valutando `e`;

- `$1`, che a ogni iterazione ha come valore `x[$i]`.

Per esempio, `iter(x,$0+$1,0)` valuta la somma degli elementi (dell'ultima dimensione) di `x`, e `iter(x,$1#$0,[])` ('#' denota l'operatore di concatenazione) produce un vettore a elementi invertiti.

Sottoespres-sioni

Pur non includendo funzioni per l'assegnazione di valore a variabili, STEL consente di identificare una o più sottoespressioni (al momento fino a 4) dell'espressione in corso di valutazione, delimitandole mediante parentesi graffe: `{`*`sottoespressione`*`}`. Tali sottoespressioni vengono valutate secondo l'ordine di valutazione dell'espressione complessiva, e il valore di ognuna di esse viene assegnato alla variabile di sistema `$w`*`i`*, *`i`*=0,...,3, e quindi può essere richiamato in punti successivi dell'espressione stessa.
Per esempio:

```
if({order(x)}==0,1,$w0==1,@x,*/@x)
```

calcola il numero di elementi dell'array `x`, secondo la logica:

- se `x` è uno scalare, cioè un array di ordine 0, produci il valore `1`; calcola nel frattempo il valore della sottoespressione `order(x)` e assegnalo alla variabile di sistema `$w0`;
- se invece `x` è un vettore, cioè un array di ordine 1, produci il valore `@x`, cioè la dimensione di `x`;
- altrimenti, cioè se `x` è una matrice o un array di dimensione superiore, produci il valore `*/@x`, cioè il prodotto delle dimensioni di `x`.

Qualche esempio

Prodotto vettoriale di due matrici x e y

```
array([numRows(x),numCols(y)],+/(x[[$i0],[]]*y[[],[
$i1]]))
```

Prodotto scalare di due vettori x e y

```
+/(x*y)
```

Range del vettore x

```
max/(x)-min/(x)
```

Media del vettore x

```
(+/x)/@x
```

Mediana del vettore x

```
if(isEven({@x}),(get({sort(x)},$w0/2-
```

```
1)+get($w1,$w0/2))/2,
get(sort(x),($w0-1)/2))
```

Deviazione standard del vettore x

```
sqrt((+/((x-mean(x))^2))/(lastDim(x)-1))
```

Vettore ottenuto rimuovendo gli 0 dal vettore x

```
iter(x,if($1!=0,$0#$1,$0),[])
```

LIUC

Il volume offre una proposta metodologica per l'insegnamento della matematica attraverso una serie di attività laboratoriali volte alla spiegazione di alcuni argomenti principali della matematica.
Il metodo proposto rientra nel contesto di una didattica delle competenze, in modo da rendere consapevole lo studente di quando applicare conoscenze e abilità, non solo a scuola ma anche nella vita quotidiana. Il ciclo di lezioni si trasforma in un vero e proprio laboratorio di attività, da affrontare singolarmente o in gruppo, così da aiutare gli studenti da una parte ad apprendere i contenuti, dall'altra a riflettere sul linguaggio stesso della matematica (qui detto "matematichese"), unendo l'apprendimento della matematica alla produzione linguistica che essa comporta. Le schede sono organizzate in due parti: nella prima parte si forniscono delle indicazioni di metodo (note didattiche, obiettivi, metodologia, strumenti), nella seconda si descrive l'attività vera e propria.

Lorella Carimali

Lorella Carimali è docente di matematica e fisica al Liceo scientifico Vittorio Veneto di Milano e collabora con CARED-Centro d'Ateneo per la Ricerca Educativo-Didattica e l'Aggiornamento della LIUC Università Cattaneo. Da anni è impegnata nell'insegnamento laboratoriale della matematica negli istituti superiori di secondo grado e nella riflessione sulla didattica per competenze. Osservatrice ed esperta INVALSI nel progetto "PON Valutazione e miglioramento per le scuole" e valutatrice progetto VSQ, è formatrice per l'Ufficio Scolastico Regionale Lombardia nei percorsi di aggiornamento del personale docente per la riprogettazione dei curricoli di matematica e dell'asse scientifico-tecnologico in termini di sviluppo di competenze. E' docente di "Analisi dei curriculi, valutazione per competenze e didattica per alunni disabili" presso l'Università degli Studi di Bergamo (TFA classe A047).

Luca Mari

Luca Mari è professore ordinario nel settore delle Misure elettriche ed elettroniche presso la Scuola di Ingegneria Industriale di LIUC Università Cattaneo e coordinatore del corso di dottorato di ricerca in Gestione Integrata d'Azienda. Ha ricoperto e ricopre incarichi di chairman nelle più importanti organizzazioni internazionali di ricerca per la metrologia (IMEKO, JCGM). E' presidente del Comitato Tecnico CEI 1/25 "Terminologia, grandezze e unità" e vicepresidente della Commissione Tecnica UNI-CEI 500 "Metrologia Generale". E' coordinatore della linea di ricerca "Metrologia" dell'Associazione Italiana "Gruppo Misure Elettriche ed Elettroniche" (GMEE). E' autore di numerose pubblicazioni scientifiche sulla scienza della misurazione su riviste internazionali e di vari libri didattici di argomento informatico.

ISBN 978-88-908806-0-5